We present a randomized algorithm for the approximate nearest neighbor problem in d-dimensional Euclidean space. Given N points $\{\boldsymbol{x}_j\}$ in $\mathbb{R}^d$, the algorithm attempts to find k nearest neighbors for each of $\boldsymbol{x}_j$, where k is a user-specified integer parameter. The algorithm is iterative, and its CPU time requirements are proportional to $T \cdot N \cdot (d \cdot (\log d) + k \cdot (\log k) \cdot (\log N)) + N \cdot k^2 \cdot (d + \log k)$, with T the number of iterations performed. The memory requirements of the procedure are of the order $N \cdot (d + k)$.

A byproduct of the scheme is a data structure, permitting a rapid search for the k nearest neighbors among $\{\boldsymbol{x}_j\}$ for an arbitrary point $\boldsymbol{x} \in \mathbb{R}^d$. The cost of each such query is proportional to $T \cdot (d \cdot (\log d) + \log(N/k) + k^2 \cdot (d + \log k))$, and the memory requirements for the requisite data structure are of the order $N \cdot (d + k) + T \cdot (d + N \cdot k)$.

The algorithm utilizes random rotations and a basic divide-and-conquer scheme, followed by a local graph search. We analyze the scheme's behavior for certain types of distributions of $\{\boldsymbol{x}_j\}$, and illustrate its performance via several numerical examples.

A Randomized Approximate Nearest Neighbors Algorithm

Peter W. Jones[†], Andrei Osipov[‡], Vladimir Rokhlin[⋆]
Research Report YALEU/DCS/RR-1434
Yale University
September 14, 2010

[†] This author's research was supported in part by the DMS grant #0602635 and the ONR grants #N000140910108, #N000140910340; [‡] this author's research was supported in part by the AFOSR grant #FA9550-09-1-02-41; [⋆] this author's research was supported in part by the ONR grant #N00014-10-1-0570 and the AFOSR grant #FA9550-09-1-02-41.

Approved for public release: distribution is unlimited.

Keywords: *Approximate nearest neighbors, randomized algorithms, fast random rotations*

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **14 SEP 2010**	2. REPORT TYPE	3. DATES COVERED **00-00-2010 to 00-00-2010**

4. TITLE AND SUBTITLE **A Randomized Approximate Nearest Neighbors Algorithm**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Yale University ,Department of Computer Science,New Haven,CT,06520**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**113**	

Contents

1 Introduction

In this paper, we describe an algorithm for finding approximate nearest neighbors (ANN) in d-dimensional Euclidean space for each of N user-specified points $\{\boldsymbol{x}_j\}$. For each point $\boldsymbol{x}_j$, the scheme produces a list of k "suspects", that have high probability of being the k closest points (nearest neighbors) in the Euclidean metric. Those of the "suspects" that are not among the "true" nearest neighbors, are close to being so.

We present several measures of performance (in terms of statistics of the k chosen suspected nearest neighbors), for different types of randomly generated data sets consisting of N points in $\mathbb{R}^d$. Unlike other ANN algorithms that have been recently proposed (see e.g. [9]), the method of this paper does not use locality-sensitive hashing. Instead we use a simple randomized divide-and-conquer approach. The basic algorithm is iterated several times, and then followed by a local graph search.

The performance of any fast ANN algorithm must deteriorate as the dimension d increases. While the running time of our algorithm only grows as $d \cdot \log d$, the statistics of the selected approximate nearest neighbors deteriorate as the dimension d increases. We provide bounds for this deterioration (both analytically and empirically), which occurs reasonably slowly as d increases. While the actual estimates are fairly complicated (see Section 4.5), it is reasonable to say that in 20 dimensions the scheme performs extremely well, and the performance does not seriously deteriorate until d is approximately 60. At $d = 100$, the degradation of the statistics displayed by the algorithm is quite noticeable.

An outline of our algorithm is as follows:

1. Choose a random rotation, acting on $\mathbb{R}^d$, and rotate the N given points.
2. Take the first coordinate, and divide the data set into two boxes, where the boxes are divided by finding the median in the first coordinate.
3. On each box from Step 2, we repeat the subdivision on the second coordinate, obtaining four boxes in total.
4. We repeat this on coordinates $3, 4$, etc., until each of the boxes has approximately k points.
5. We do a local search on the tree of boxes to obtain approximately k "suspects", for each point $\boldsymbol{x}_j$.
6. The above procedure is iterated T times, and for each point $\boldsymbol{x}_j$, we select from the $T \cdot k$ "suspects" the k closest discovered points for $\boldsymbol{x}_j$.
7. Perform a local graph search on the collections of suspects, obtained in Step 6 (we call this local graph search "supercharging"). Among k^2 "candidates" obtained from the local graph search, we select the best k points and declare these "the suspected approximate nearest neighbors", or "suspects".

The data structure generated by this algorithm allows one to find, for a new data point $\boldsymbol{y}$, the k suspected approximate nearest neighbors in the original dataset. This search is quite rapid, as we need only follow the already generated tree structure of the boxes, obtained in the steps listed above. One can easily see that the depth of the binary tree, generated by Steps 1 through 4, is $\log_2(N/k)$. This means that we can use the T trees generated, and then pass to Step 7 (see Sections 4.2.4, 4.3.4).

Almost all known techniques for solving ANN problems use tree structures (see e.g. [5], [9]). Two apparently novel features of our method are the use of fast random rotations (Step 1), and the local graph search (Step 7), which dramatically increases the accuracy of the scheme. We use the Fast Fourier Transform to generate our random rotations, and this accounts for the factor of $\log d$ that appears in the running time (see Section 2.4 for details). Our use of random rotations replaces the usual projection argument used in other ANN algorithms, where one projects the data on a random subspace. As far as we know, the use of fast rotations for applications of this type appears first in [2] (see [3] and the references therein for a brief history). The use of random rotations (as in our paper) or random projections (as used elsewhere in ANN algorithms) takes advantage of the same underlying phenomenon; namely the Johnson-Lindenstrauss Lemma. (The JL Lemma roughly states that projection of N points on a random subspace of dimension $C(\varepsilon) \cdot (\log N)$ has expected distortion $1+\varepsilon$, see e.g. [10].) We have chosen to use random rotations in place of the usual random projections generated by selecting random Gaussian vectors. The fast random rotations require $O(d \cdot (\log d))$ operations, which is an improvement over methods using random projections (see [13], [14]).

The $N \times k$ lookup table arising in Step 7 is the adjacency matrix of a graph whose vertices are the points $\{\boldsymbol{x}_j\}$. In Step 7 we perform a depth one search on this graph, and obtain $\leq k + k^2$ "candidates" (of whom we select the "suspects"). This accounts for the factor of k^2 in the running time. Due to degradation of the running time, we have chosen not to perform searches of depth greater than one.

The algorithm has been tested on a number of artificially generated point distributions. Results of some of those tests are presented in Section 5 below.

The paper is organized as follows. In Section 2, we summarize the mathematical and numerical facts to be used in subsequent sections. In Section 3, we develop the analytical apparatus to be used in the analysis of the algorithm in the case, where the points $\{\boldsymbol{x}_j\}$ are distributed according to the Gaussian law. In Section 4, we describe the Randomized Approximate Nearest Neighbors algorithm (RANN) and analyze its cost and performance. In Section 5, we illustrate the performance of the algorithm with several numerical examples. In Section 6, we discuss possible generalizations and modifications of the algorithm.

2 Mathematical Preliminaries

In this section, we introduce notation and summarize several well known facts to be used in the rest of the paper.

2.1 Euclidean Space

Suppose that $d > 0$ is a positive integer. We denote by $\mathbb{R}^d$ the d-dimensional linear Euclidean space. The vectors in $\mathbb{R}^d$ are denoted by bold lower case letters, e.g.

$$\boldsymbol{x} = (x(1), \ldots, x(d)) \,. \tag{1}$$

We denote the Euclidean (or l^2) norm of $\boldsymbol{x}$ by

$$\|\boldsymbol{x}\| = \|\boldsymbol{x}\|_2 = \sqrt{x(1)^2 + \cdots + x(d)^2}. \tag{2}$$

Suppose that $B = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ is a collection of N points in $\mathbb{R}^d$ and that an integer i is between 1 and N. We denote by $\boldsymbol{x}_{t(i,j)}$ the jth nearest neighbor of $\boldsymbol{x}_i$. For a subset A of B, we denote by $\boldsymbol{x}_{t(i,j,A)}$ the jth nearest neighbor of $\boldsymbol{x}_i$ in A.

Suppose that $d \geq L > 0$ are positive integers. Suppose further that

$$\boldsymbol{\sigma} = \sigma_1 \dots \sigma_L, \quad \boldsymbol{\mu} = \mu_1 \dots \mu_L, \quad \sigma_i, \mu_j \in \{+, -\} \tag{3}$$

are two words of symbols $+, -$ of length L. We define the degree of contact $Con(\boldsymbol{\sigma}, \boldsymbol{\mu})$ between $\boldsymbol{\sigma}$ and $\boldsymbol{\mu}$ to be the number of positions at which the corresponding symbols are different. In other words,

$$Con(\boldsymbol{\sigma}, \boldsymbol{\mu}) = \left|\{i \ : \ 1 \leq i \leq L, \quad \sigma_i \neq \mu_i\}\right|. \tag{4}$$

The following definition illustrates the concept of degree of contact.

Definition 1. *Suppose that $L > 0$ is a positive integer, and that $\boldsymbol{\sigma}$ is a word of symbols $+, -$ of length L, as in (3). We define $\boldsymbol{\sigma}^0 = \boldsymbol{\sigma}$, and, for $j = 1, \dots, L$, we define $\boldsymbol{\sigma}^j$ to be the word obtained by altering the jth symbol in $\boldsymbol{\sigma}$, and leaving the others unchanged. In other words, for all $i = 1, \dots, L$,*

$$\boldsymbol{\sigma}^j_i = \begin{cases} \sigma_i, & i \neq j, \\ +, & i = j, \ \ \sigma_i = -, \\ -, & i = j, \ \ \sigma_i = +. \end{cases} \tag{5}$$

The words $\boldsymbol{\sigma}^0, \dots, \boldsymbol{\sigma}^L$ are precisely those words, whose degree of contact with $\boldsymbol{\sigma}$ is either zero or one.

In a mild abuse of notation, we say that two disjoint sets $A_{\boldsymbol{\sigma}}$ and $A_{\boldsymbol{\mu}}$ (or their elements) have degree of contact j if $Con(\boldsymbol{\sigma}, \boldsymbol{\mu}) = j$. For example, x and y have degree of contact 1 if $x \in A_{\boldsymbol{\sigma}}$, $y \in A_{\boldsymbol{\mu}}$ and $\boldsymbol{\sigma}, \boldsymbol{\mu}$ differ at precisely one symbol. We define the subset $Q^d_{\boldsymbol{\sigma}}$ of $\mathbb{R}^d$ by the formula

$$Q^d_{\boldsymbol{\sigma}} = \left\{\boldsymbol{x} \in \mathbb{R}^d \ : \ \operatorname{sgn}(x(i)) = \sigma_i, \quad i = 1, \dots, L\right\}. \tag{6}$$

In other words, a vector $\boldsymbol{x} \in \mathbb{R}^d$ belongs to $Q^d_{\boldsymbol{\sigma}}$ if and only if the signs of its first L coordinates coincide with the corresponding symbols of the word $\boldsymbol{\sigma}$.

2.2 Analysis

In this section, we summarize some well known facts from the real and complex analysis. These facts can be found in [1], [11], [15], [16].

Suppose that $x > 0$ is a positive real number. In agreement with the standard practice, we define the real gamma function by the formula

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} \, dt. \tag{7}$$

Suppose that $d > 1$ is an integer. The d-dimensional volume of the d-dimensional unit ball

$$B_d((0,\dots,0),1) = \left\{ \boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\| \le 1 \right\} \tag{8}$$

is given by the formula

$$\mathrm{Vol}\,(d) = \frac{\pi^{d/2}}{\Gamma\,(d/2+1)}. \tag{9}$$

The $(d-1)$-dimensional area of the hypersphere $\partial B_d((0,\dots,0),1)$ is given by the formula

$$\mathrm{Area}\,(d) = d \cdot \mathrm{Vol}\,(d) = \frac{d \cdot \pi^{d/2}}{\Gamma\,(d/2+1)}. \tag{10}$$

We denote the positive part of the d-dimensional hypersphere of radius $r > 0$ by

$$S_d^+(r) = \left\{ \boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\| = r,\;\; x(j) > 0,\;\; j = 1,\dots,d \right\}. \tag{11}$$

The error function is an entire function $\mathbb{C} \to \mathbb{C}$, defined by the formula

$$\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2}\, dt. \tag{12}$$

The complementary error function is an entire function $\mathbb{C} \to \mathbb{C}$, defined by the formula

$$\mathrm{erfc}(z) = \frac{2}{\sqrt{\pi}} \int_z^\infty e^{-t^2}\, dt. \tag{13}$$

For all complex $z \in \mathbb{C}$,

$$\mathrm{erf}(z) + \mathrm{erfc}(z) = 1. \tag{14}$$

Suppose that $f : \mathbb{R} \to \mathbb{C}$ is a continuous function in $L^1(\mathbb{R})$ (that is, f is defined on the real axis and is absolutely integrable). We define its Fourier transform $h_f : \mathbb{R} \to \mathbb{C}$ by the formula

$$h_f(x) = \int_{-\infty}^\infty f(t) \cdot e^{ixt}\, dt. \tag{15}$$

If the function h_f itself belongs to $L^1(\mathbb{R})$, then the Fourier inversion formula holds, that is, for all real t

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^\infty h_f(x) \cdot e^{-ixt}\, dx. \tag{16}$$

Definition 2 (average)**.** *Suppose that $d > 0$ is a positive integer, $Q \subseteq \mathbb{R}^d$ is a subset of $\mathbb{R}^d$, and $f : Q \to \mathbb{R}$ is a function. Suppose further, that the integral of f over Q is well defined and finite, and also*

$$0 < \int_Q 1 < \infty. \tag{17}$$

We define the average of f over Q by the formula

$$\operatorname*{Avg}_Q(f) = \frac{\int_Q f}{\int_Q 1}. \tag{18}$$

2.3 Probability

In this section, we summarize some well known facts from the probability theory. These facts can be found in [1], [6], [7], [8].

We say that the discrete random variable X has binomial distribution $Bin(N, p)$ with integer parameter $N > 0$ and real parameter $0 < p < 1$, if for all integer $k = 1, \dots, N$ the probability that X equals to k is given by the formula

$$\mathbb{P}\,\{X = k\} = \binom{N}{k} \cdot p^k \cdot (1-p)^{N-k}. \tag{19}$$

The binomial distribution describes the sum of N independent identically distributed (i.i.d.) random variables, that take value 1 with probability p and value 0 with probability $1 - p$. Its expectation and variance are given by the formulae

$$\mathbb{E}\left[Bin(N, p)\right] = N \cdot p, \quad \operatorname{Var}\left[Bin(N, p)\right] = N \cdot p \cdot (1-p). \tag{20}$$

The one-dimensional standard normal distribution $N(0, 1)$ with mean zero and standard deviation one is defined by its probability density function (pdf)

$$f_{N(0,1)}(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2}, \quad -\infty < t < \infty. \tag{21}$$

Its cumulative distribution function (cdf) is given by the formula

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2}\,dt = \frac{1}{2} \cdot \left(1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right). \tag{22}$$

Suppose that $d > 0$ is a positive integer. We say that the random vector $\boldsymbol{x}$ has standard normal d-dimensional distribution $N(0_d, I_d)$, if all of its coordinates are independent standard normal random variables.

Suppose now that $\boldsymbol{x} \sim N(0_d, I_d)$. Then $\|\boldsymbol{x}\|^2$ has distribution[1] χ^2_d with pdf

$$f_{\chi^2_d}(t) = \frac{t^{d/2-1} \cdot e^{-t/2}}{2^{d/2} \cdot \Gamma(d/2)}, \quad t > 0. \tag{23}$$

Its expectation and variance are given by the formulae

$$\mathbb{E}\left[\chi^2_d\right] = d, \quad \operatorname{Var}\left[\chi^2_d\right] = 2d. \tag{24}$$

Also, if $\boldsymbol{a} \in \mathbb{R}^d$ is a fixed vector and we denote $\lambda = \|\boldsymbol{a}\|^2$, the random variable $\|\boldsymbol{x} - \boldsymbol{a}\|^2$ has distribution[2] $\chi^2(d, \lambda)$ with pdf

$$f_{\chi^2(d,\lambda)}(t) = e^{-\lambda/2} \sum_{j=0}^{\infty} \frac{(\lambda/2)^j}{j!} f_{\chi^2_{d+2j}}(t), \quad t > 0. \tag{25}$$

[1] Chi-square with d degrees of freedom.

[2] Noncentral chi-square with d degrees of freedom and noncentrality parameter λ.

Its expectation and variance are given by the formulae

$$\mathbb{E}\left[\chi^2(d,\lambda)\right] = d + \lambda, \quad \operatorname{Var}\left[\chi^2(d,\lambda)\right] = 2\left(d + 2\lambda\right). \tag{26}$$

The Fourier transform of the pdf of $\chi^2(d,\lambda)$ for all real x is given by the formula

$$\int_{-\infty}^{\infty} f_{\chi^2(d,\lambda)}(t) \cdot e^{ixt}\, dt = \exp\left[\frac{i\lambda x}{1-2ix}\right] \cdot \left(\frac{1}{\sqrt{1-2ix}}\right)^d, \tag{27}$$

where the principal branch of the complex square root is taken.

The beta distribution $B(\alpha,\beta)$ with shape parameters $\alpha, \beta > 0$ is defined by its pdf

$$f_{B(\alpha,\beta)}(t) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \cdot t^{\alpha-1}\left(1-t\right)^{\beta-1}, \quad 0 < t < 1. \tag{28}$$

Its expectation and variance are given by the formulae

$$\mathbb{E}\left[B(\alpha,\beta)\right] = \frac{\alpha}{\alpha+\beta}, \quad \operatorname{Var}\left[B(\alpha,\beta)\right] = \frac{\alpha\beta}{\left(\alpha+\beta\right)^2\left(\alpha+\beta+1\right)}. \tag{29}$$

Definition 3 (order statistic)**.** *Suppose that $N > 0$ is a positive integer, and that $X_1, \dots, X_N$ is a sequence of real random variables. Suppose further that i is an integer between 1 and N. The ith order statistic $X_{(i)}$ is the random variable, obtained by taking the ith smallest value among $X_1, \dots, X_N$. In other words, we choose a permutation $\pi = \pi(X_1, \dots, X_N)$ of the numbers $\{1, \dots, N\}$, such that*

$$X_{\pi(1)} \le X_{\pi(2)} \le \cdots \le X_{\pi(N)}, \tag{30}$$

and then define

$$X_{(i)} = X_{\pi(i)}. \tag{31}$$

The following theorem illustrates Definition 3 by providing an elementary bound on the expectation of order statistics for positive random variables.

Theorem 1. *Suppose that $N > 0$ is a positive integer, and that $X_1, \dots, X_N$ is a sequence of i.i.d. real positive random variables with expectation μ. Then, for any $i = 1, \dots, N$,*

$$\mathbb{E}\left[X_{(i)}\right] \le \frac{N}{N+1-i} \cdot \mu. \tag{32}$$

Proof. By contradiction, suppose that (32) does not hold. In other words, for some i,

$$\mathbb{E}\left[X_{(i)}\right] > \frac{N}{N+1-i} \cdot \mu. \tag{33}$$

Obviously, for all $j = i, \dots, N$,

$$\mathbb{E}\left[X_{(i)}\right] \le \mathbb{E}\left[X_{(j)}\right]. \tag{34}$$

Also, due to linearity of the expectation,

$$\sum_{j=1}^{N} \mathbb{E}\left[X_{(j)}\right] = \sum_{j=1}^{N} \mathbb{E}\left[X_j\right] = N \cdot \mu. \tag{35}$$

We combine (30), (31), (33), (34) and (35) to conclude that

$$N \cdot \mu \geq \sum_{j=i}^{N} \mathbb{E}\left[X_{(j)}\right] \geq (N-i+1) \cdot \mathbb{E}\left[X_{(i)}\right] > N \cdot \mu, \tag{36}$$

in contradiction to non-negativity of μ. ■

The following well known theorem describes the distribution of order statistics of uniform random variables.

Theorem 2. *Suppose that $N > 0$ is a positive integer, and that $U_1, \dots, U_N$ are i.i.d. uniform random variables in $(0,1)$. Then, for any $i = 1, \dots, N$, the distribution of the order statistic $U_{(i)}$ is given by the formula*

$$U_{(i)} \sim B(i, N+1-i), \tag{37}$$

where $B(i, N+1-i)$ is the beta distribution, whose pdf is defined via (28).

Definition 4 (random vector conditioned on a set)**.** *Suppose that $d > 0$ is a positive integer, and that $\boldsymbol{x}$ is a real random d-dimensional vector with pdf $f_{\boldsymbol{x}}$. Suppose further that Q is a subset of $\mathbb{R}^d$ and that the probability that $\boldsymbol{x}$ is in Q is positive. In other words,*

$$\mathbb{P}\left\{\boldsymbol{x} \in Q\right\} = \int_Q f_{\boldsymbol{x}}(\boldsymbol{y})\, d\boldsymbol{y} > 0. \tag{38}$$

The random vector $\boldsymbol{x}|Q$ ($\boldsymbol{x}$ conditioned on Q) is defined by its pdf

$$f_{\boldsymbol{x}|Q}(\boldsymbol{y}) = \begin{cases} f_{\boldsymbol{x}}(\boldsymbol{y}) / \mathbb{P}\left\{\boldsymbol{x} \in Q\right\} & \boldsymbol{y} \in Q, \\ 0 & \boldsymbol{y} \notin Q. \end{cases} \tag{39}$$

For example, if $X \sim N(0,1)$, then $|X|$ is the standard normal variable conditioned on $(0, \infty)$. Moreover, its pdf is given by the formula

$$f_{|X|}(t) = f_{N(0,1)|(0,\infty)}(t) = \sqrt{\frac{2}{\pi}} \cdot e^{-t^2/2}, \quad t > 0. \tag{40}$$

Definition 5 (bounded in probability)**.** *Suppose that $Y_1, Y_2, \dots$ is a sequence of real random variables. We say that this sequence is bounded in probability, written as*

$$Y_N = O_p\left(1\right), \tag{41}$$

if for any $\varepsilon > 0$ there exists $M(\varepsilon) > 0$ such that for all integer $N > 0$,

$$\mathbb{P}\left\{|Y_N| > M(\varepsilon)\right\} \leq \varepsilon. \tag{42}$$

Definition 6 (big-O in probability)**.** *Suppose that $Y_1, Y_2, \ldots$ is a sequence of real random variables, and that $a_1, a_2, \ldots$ is a sequence of non-zero real numbers. We say that*

$$Y_N = O_p\left(a_N\right), \tag{43}$$

if the sequence $\{Y_N/a_N\}_{N=1}^{\infty}$ is bounded in probability, in other words,

$$\frac{Y_N}{a_N} = O_p\left(1\right). \tag{44}$$

For example, suppose that $Y_1, Y_2, \ldots$ is a sequence of i.i.d. random variables with mean μ and standard deviation one. Then, due to the central limit theorem,

$$\sum_{k=1}^{N} Y_k = N \cdot \mu + O_p(\sqrt{N}) = N \cdot \left(\mu + O_p\left(\frac{1}{\sqrt{N}}\right)\right), \tag{45}$$

in the sense of Definition 6.

Theorem 3. *Suppose that μ and $\sigma > 0$ are real numbers, and $X_1, X_2, \ldots$ is a sequence of i.i.d. real random variables. Suppose further, that $g : \mathbb{R} \to \mathbb{R}$ is a continuously differentiable function, and that*

$$g'(\mu) \neq 0. \tag{46}$$

Suppose also that

$$\overset{\mathbb{D}}{\lim_{k\to\infty}} \sqrt{k} \cdot (X_k - \mu) = N(0, \sigma^2), \tag{47}$$

in other words, the sequence of X_k, shifted by μ and rescaled by $\sqrt{k}$, converges in distribution to $N(0, \sigma^2)$. Then,

$$\overset{\mathbb{D}}{\lim_{k\to\infty}} \sqrt{k} \cdot (g(X_k) - g(\mu)) = N(0, \sigma^2 \cdot \left(g'(\mu)\right)^2), \tag{48}$$

and, in addition, the expectation of $g(X_k)$ is given by the formula

$$\mathbb{E}\left[g\left(X_k\right)\right] = g(\mu) + O\left(\frac{1}{\sqrt{k}}\right). \tag{49}$$

Definition 7 (empirical distribution function)**.** *Suppose that $N > 0$ is a positive integer, and $X_1, \ldots, X_N$ are i.i.d. real continuous random variables with common cdf $F : \mathbb{R} \to [0, 1]$. For all real t, we define the random variable $\hat{F}_N(t)$ by the formula*

$$\hat{F}_N(t) = \frac{1}{N} \cdot \left|\{X_i \;:\; X_i \leq t, \;\; i = 1, \ldots, N\}\right|. \tag{50}$$

In other words, $\hat{F}_N(t)$ is the proportion of those X_i's, whose values are less than or equal to t. The random function $\hat{F}_N : \mathbb{R} \to [0, 1]$ is called the empirical distribution function.

The following theorem describes some elementary asymptotical properties of the empirical distribution function.

Theorem 4. *Suppose that $\hat{F}_N$ is defined via* (50)*. Then, for all real t,*

$$\hat{F}_N(t) \sim \frac{1}{N} \cdot Bin\left(N, F(t)\right). \tag{51}$$

In other words, $N \cdot \hat{F}_N(t)$ has binomial distribution with parameters N and $p = F(t)$, as in (19)*,* (20)*. In particular,*

$$\overset{\mathbb{D}}{\lim_{N \to \infty}} \sqrt{N} \cdot \left(\hat{F}_N(t) - F(t)\right) = N\left(0, F(t) \cdot (1 - F(t))\right). \tag{52}$$

In other words, $\hat{F}_N(t)$, shifted by $F(t)$ and rescaled by $\sqrt{N}$, converges in distribution to the normal distribution with mean zero and variance $F(t) \cdot (1 - F(t))$.

2.4 Pseudorandom orthogonal transformations

In this section, we describe a fast method (presented in [13], [14]) for the generation of random orthogonal transformations and their application to arbitrary vectors.

Suppose that $d, M_1, M_2 > 0$ are positive integers. We define a pseudorandom d-dimensional orthogonal transformation Θ as a composition of $M_1 + M_2 + 1$ linear operators

$$\Theta = \left(\prod_{j=1}^{M_1} Q_j^{(d)} P_j^{(d)}\right) \cdot F^{(d)} \cdot \left(\prod_{j=M_1+1}^{M_1+M_2} Q_j^{(d)} P_j^{(d)}\right). \tag{53}$$

The linear operators $P_j^{(d)} : \mathbb{R}^d \to \mathbb{R}^d$ with $j = 1, \dots, M_1 + M_2$ are defined in the following manner. We generate permutations $\pi_1, \dots, \pi_{M_1+M_2}$ of the numbers $\{1, \dots, d\}$, uniformly at random and independent of each other. Then for all $\boldsymbol{x} \in \mathbb{R}^d$, we define $P_j^{(d)} \boldsymbol{x}$ by the formula

$$\left(P_j^{(d)} \boldsymbol{x}\right)(i) = x(\pi_j(i)), \quad i = 1, \dots, d. \tag{54}$$

In other words, $P_j^{(d)}$ permutes the coordinates of the vector $\boldsymbol{x}$ according to π_j. $P_j^{(d)}$ can be represented by a $d \times d$ matrix P_j, defined by the formula

$$P_j(k, l) = \begin{cases} 1 & l = \pi_j(k), \\ 0 & l \neq \pi_j(k), \end{cases} \tag{55}$$

for $k, l = 1, \dots, d$. Obviously, the operators $P_j^{(d)}$ are orthogonal.

The linear operators $Q_j^{(d)} : \mathbb{R}^d \to \mathbb{R}^d$ with $j = 1, \dots, M_1 + M_2$ are defined as follows. We construct $(d-1) \cdot (M_1 + M_2)$ independent pseudorandom numbers, $\theta_j(1), \dots, \theta_j(d-1)$ with $j = 1, \dots, M_1 + M_2$, uniformly distributed in $(0, 2\pi)$. Then we define the auxiliary linear operator $Q_{j,k} : \mathbb{R}^d \to \mathbb{R}^d$ for $k = 1, \dots, d-1$ by the formula

$$\left(Q_{j,k}(\boldsymbol{x})\right)(i) = \begin{cases} \cos(\theta_j(k)) \cdot x(k) + \sin(\theta_j(k)) \cdot x(k+1), & i = k, \\ -\sin(\theta_j(k)) \cdot x(k) + \cos(\theta_j(k)) \cdot x(k+1), & i = k+1, \\ x(i) & i \notin \{k, k+1\}, \end{cases} \tag{56}$$

for all $\boldsymbol{x} \in \mathbb{R}^d$. In other words,

$$(Q_{j,k}(\boldsymbol{x})) \begin{pmatrix} k \\ k+1 \end{pmatrix} = \begin{pmatrix} \cos(\theta_j(k)) & \sin(\theta_j(k)) \\ -\sin(\theta_j(k)) & \cos(\theta_j(k)) \end{pmatrix} \cdot \begin{pmatrix} x(k) \\ x(k+1) \end{pmatrix}, \tag{57}$$

and the rest of the coordinates of $Q_{j,k}(\boldsymbol{x})$ coincide with those of $\boldsymbol{x}$. We define $Q_j^{(d)}$ by the formula

$$Q_j^{(d)} = Q_{j,d-1} \cdot Q_{j,d-2} \cdot \dots \cdot Q_{j,1}. \tag{58}$$

Obviously, the operators $Q_j^{(d)}$ are orthogonal.

The linear operator $F^{(d)} : \mathbb{R}^d \to \mathbb{R}^d$ is defined as follows. First suppose that d is even and that $d_2 = d/2$. We define the $d_2 \times d_2$ discrete Fourier transform matrix T by the formula

$$T(k,l) = \frac{1}{\sqrt{d_2}} \cdot \exp\left[-\frac{2\pi i(k-1)(l-1)}{d_2}\right], \tag{59}$$

where $k, l = 1, \dots, d_2$ and $i = \sqrt{-1}$. The matrix T represents a unitary operator $\mathbb{C}^{d_2} \to \mathbb{C}^{d_2}$. We then define the one-to-one linear operator $Z : \mathbb{R}^d \to \mathbb{C}^{d_2}$ by the formula

$$Z\boldsymbol{x} = \begin{pmatrix} x(1) + i \cdot x(2), \\ x(3) + i \cdot x(4), \\ \dots \\ x(2d_2 - 1) + i \cdot x(2d_2) \end{pmatrix} \tag{60}$$

for all $\boldsymbol{x} \in \mathbb{R}^d$. Eventually, we define $F^{(d)}$ by the formula

$$F^{(d)} = Z^{-1} \cdot T \cdot Z \tag{61}$$

for even d. If d is odd, we define $F^{(d)}\boldsymbol{x}$ for all $\boldsymbol{x} \in \mathbb{R}^d$ by applying $F^{(d-1)}$ to the first $d-1$ coordinates of $\boldsymbol{x}$ and leaving its last coordinate unchanged. Obviously, the operators T, Z, defined by (59), (60), respectively, preserve the norm of any vector $\boldsymbol{x} \in \mathbb{R}^d$. Therefore, $F^{(d)}$ is a real orthogonal transformation $\mathbb{R}^d \to \mathbb{R}^d$.

The cost of the generation of a random permutation (see e.g. [12]) is $O(d)$ operations. The cost of the application of each $P_j^{(d)}$ to a vector $\boldsymbol{x} \in \mathbb{R}^d$ is obviously d operations due to (54).

The cost of generation of $d-1$ uniform random variables is $O(d)$ operations. Also, the cost of application of each $Q_j^{(d)}$ to a vector $\boldsymbol{x} \in \mathbb{R}^d$ is $O(d)$ operations due to (56), (58).

Finally, the cost of the fast discrete Fourier transform is $O(d \cdot \log d)$ operations, and the cost of the application of $F^{(d)}$ to a vector $\boldsymbol{x} \in \mathbb{R}^d$ is $O(d \cdot \log d)$ operations due to (59), (60) and (61).

Thus the cost of the generation of Θ defined via (53) is

$$\text{Cost}(\Theta) = O\left(d \cdot (M_1 + M_2 + \log d)\right). \tag{62}$$

Moreover, the cost of application of Θ to a vector $\boldsymbol{x} \in \mathbb{R}^d$ is also given by the formula (62).

Remark 1. *The use of the Hadamard matrix (without 2×2 rotations) appears in a related problem studied by Ailon and Liberty [4].*

3 Analytical Apparatus

The purpose of this section is to provide the analytical apparatus to be used in the rest of the paper.

The following theorem generalizes Theorem 2 in Section 2.3. Its proof is provided here for the sake of completeness.

Theorem 5. *Suppose that $N > 0$ is a positive integer, and that $X_1, \dots, X_N$ are i.i.d. real continuous random variables with cdf $F : \mathbb{R} \to (0,1)$. Then, for any $i = 1, \dots, N$, the expectation of the order statistic $X_{(i)}$ (see Definition 3 in Section 2.3) is given by the formula*

$$\mathbb{E}\left[X_{(i)}\right] = \int_0^1 F^{-1}(t) \cdot f_{B(i,N+1-i)}(t)\, dt, \tag{63}$$

where $f_{B(i,N+1-i)}$ is the pdf of the beta distribution, defined via (28).

Proof. For the purpose of proving (63), we introduce N random variables $U_1, \dots, U_N$, defined by the formula

$$U_i = F\left(X_i\right), \tag{64}$$

for $i = 1, \dots, N$. Obviously, U_i are i.i.d. uniform random variables in $(0,1)$. Since the function F in (64) is monotonically increasing, it preserves order. In other words, the order statistic $U_{(i)}$ satisfies the formula

$$U_{(i)} = F\left(Y_{(i)}\right), \tag{65}$$

for all $i = 1, \dots, N$. Due to Theorem 2 in Section 2.3,

$$U_{(i)} \sim B(i, N+1-i), \tag{66}$$

for all $i = 1, \dots, N$, where $B(i, N+1-i)$ is the beta distribution, whose pdf is defined via (28). Thus the identity (63) follows from the combination of (28), (64), (65) and (66). ■

Theorem 6. *Suppose that $c > 1$ is a positive real number, and that the function $F : \mathbb{R} \to (0,1)$ is the cdf of a real continuous random variable. Suppose further, that the inverse function $F^{-1} : (0,1) \to \mathbb{R}$ is continuously differentiable, and that*

$$\left(F^{-1}\right)' \left(\frac{1}{c}\right) \neq 0. \tag{67}$$

For every positive integer $k > 0$ such that $N = c \cdot k - 1$ is an integer, we consider the sequence of i.i.d. random variables $X_1, \dots, X_N$ with cdf F, and define by $X_{(k)}$ the kth order statistic of this sequence, as in Definition 3 in Section 2.3. Then,

$$\mathbb{E}\left[X_{(k)}\right] = F^{-1}\left(\frac{1}{c}\right) + O\left(\frac{1}{\sqrt{k}}\right), \quad k \to \infty. \tag{68}$$

In other words, up to an error of order $O(k^{-1/2})$, the expectation of $X_{(k)}$ is given by $F^{-1}\left(k/(N+1)\right)$.

Proof. Consider the random variables

$$Y_k = B(k, N+1-k) = B(k, (c-1)\cdot k), \tag{69}$$

where $B(k, N+1-k)$ is the beta distribution, whose pdf is defined via (28). Due to (29),

$$\begin{aligned}\mathbb{E}[Y_k] &= \frac{k}{N+1} = \frac{1}{c},\\ \operatorname{Var}[Y_k] &= \frac{k}{(N+1)^2}\cdot\frac{N+1-k}{N+2} = \frac{c-1}{c^2}\cdot\frac{1}{ck+1}.\end{aligned} \tag{70}$$

Therefore,

$$\overset{\mathbb{D}}{\lim_{k\to\infty}} \sqrt{k}\cdot\left(Y_k - \frac{1}{c}\right) = N\left(0, \frac{c-1}{c^3}\right), \tag{71}$$

similar to (47). Thus the identity (68) follows from the combination of (71), Theorem 3 in Section 2.3 and (64), (65), (66) in the proof of Theorem 5. ■

Theorem 7. *Suppose that $a > 0$ is a positive real number, and that $X \sim N(0,1)$. We define the random variable D_a^- by the formula*

$$D_a^- = |(-|X|) - a|^2 = ||X| + a|^2. \tag{72}$$

In other words, D_a^- is the square of the distance from a to the standard normal variable conditioned on $(-\infty, 0)$ (see Definition 4 in Section 2.3). Then the pdf of D_a^- is given by the formula

$$f_{D_a^-}(t) = \frac{1}{\sqrt{2\pi t}}\cdot\chi_{(a^2,\infty)}(t)\cdot e^{-\frac{(a-\sqrt{t})^2}{2}}. \tag{73}$$

Moreover, the Fourier transform $h_a^- : \mathbb{R} \to \mathbb{C}$ of $f_{D_a^-}$ is given by the formula

$$\begin{aligned}h_a^-(x) &= \int_{-\infty}^{\infty} f_{D_a^-}(t)\cdot e^{ixt}\,dt\\ &= \exp\left[\frac{ia^2x}{1-2ix}\right]\cdot\operatorname{erfc}\left[-\frac{iax\sqrt{2}}{\sqrt{1-2ix}}\right]\cdot\frac{1}{\sqrt{1-2ix}},\end{aligned} \tag{74}$$

where the principal branch of the complex square root is taken.

Proof. Suppose that $t > 0$ is a real positive number. Clearly, $D_a^- > a^2$ with probability one, so we may further assume that $t > a^2$ and evaluate the cdf of D_a^- at t by computing the probability of D_a^- being smaller than t to obtain

$$\begin{aligned}F_{D_a^-}(t) &= \int_{a^2}^{t} f_{D_a^-}(s)\,ds\\ &= \mathbb{P}\left\{|-|X|-a|^2 < t\right\} = \mathbb{P}\left\{||X|+a| < \sqrt{t}\right\}\\ &= \mathbb{P}\left\{-\sqrt{t}-a < |X| < \sqrt{t}-a\right\} = F_{|X|}(\sqrt{t}-a),\end{aligned} \tag{75}$$

where $F_{|X|}$ is the cdf of $|X|$. We recall that the pdf of $|X|$ is given by (40) in Section 2.3 and differentiate (75) with respect to t to obtain (73). To demonstrate (74), we define w by the formula

$$w^2 = \frac{1-2ix}{2} \tag{76}$$

and perform the change of variables $s^2 = t$ to compute

$$\begin{aligned}
&\int_{a^2}^{\infty} \frac{e^{ixt}}{\sqrt{2\pi t}} \cdot e^{-(a-\sqrt{t})^2/2}\, dt = \\
&\sqrt{\frac{2}{\pi}} \int_a^{\infty} e^{ixs^2} \cdot e^{-(a-s)^2/2}\, ds = \\
&\exp\left[-\frac{a^2}{2} + \frac{a^2}{4w^2}\right] \cdot \sqrt{\frac{2}{\pi}} \int_a^{\infty} e^{-(sw-a/2w)^2}\, ds = \\
&\exp\left[\frac{a^2}{2}\left(\frac{1}{1-2ix} - 1\right)\right] \cdot \sqrt{\frac{2}{\pi}} \int_{aw-a/2w}^{\infty} e^{-z^2}\, \frac{dz}{w} = \\
&\exp\left[\frac{ia^2x}{1-2ix}\right] \cdot \frac{\sqrt{2}}{\sqrt{1-2ix}} \cdot \sqrt{\frac{2}{\pi}} \int_{a(2w^2-1)/2w}^{\infty} e^{-z^2}\, dz = \\
&\exp\left[\frac{ia^2x}{1-2ix}\right] \cdot \frac{1}{\sqrt{1-2ix}} \cdot \operatorname{erfc}\left[-\frac{iax\sqrt{2}}{\sqrt{1-2ix}}\right],
\end{aligned} \tag{77}$$

which establishes (74). ■

Theorem 8. *Suppose that $a > 0$ is a positive real number, and that $X \sim N(0,1)$. We define the random variable D_a^+ by the formula*

$$D_a^+ = ||X| - a|^2. \tag{78}$$

In other words, D_a^+ is the square of the distance from a to the standard normal variable conditioned on $(0,\infty)$ (see Definition 4 in Section 2.3). Then the pdf of D_a^+ is given by the formula

$$f_{D_a^+}(t) = \frac{1}{\sqrt{2\pi t}}\left(e^{-\frac{(a+\sqrt{t})^2}{2}} + \chi_{(0,a^2)}(t) \cdot e^{-\frac{(a-\sqrt{t})^2}{2}}\right). \tag{79}$$

Moreover, the Fourier transform $h_a^+ : \mathbb{R} \to \mathbb{C}$ of $f_{D_a^+}$ is given by the formula

$$\begin{aligned}
h_a^+(x) &= \int_{-\infty}^{\infty} f_{D_a^+}(t) \cdot e^{ixt}\, dt \\
&= \exp\left[\frac{ia^2x}{1-2ix}\right] \cdot \frac{2}{\sqrt{1-2ix}} - h_a^-(x),
\end{aligned} \tag{80}$$

where $h_a^-(x)$ is defined by (74), and the principal branch of the complex square root is taken.

Proof. Suppose that $t > 0$ is a real positive number. We evaluate the cdf of D_a^+ at t by computing the probability of D_a^+ being smaller than t to obtain

$$\begin{aligned} F_{D_a^+}(t) &= \int_0^t f_{D_a^+}(s)\, ds \\ &= \mathbb{P}\left\{ \left||X| - a\right|^2 < t \right\} = \mathbb{P}\left\{ a - \sqrt{t} < |X| < a + \sqrt{t} \right\} \\ &= \begin{cases} F_{|X|}(a+\sqrt{t}) - F_{|X|}(a-\sqrt{t}) & \text{if } \sqrt{t} \le a, \\ F_{|X|}(a+\sqrt{t}) & \text{if } \sqrt{t} > a, \end{cases} \end{aligned} \tag{81}$$

where $F_{|X|}$ is the cdf of $|X|$. We recall that the pdf of $|X|$ is given by (40) and differentiate (81) with respect to t to obtain (79). Next, due to (72) and (78),

$$|X - a|^2 = \begin{cases} D_a^- & \text{with probability } 1/2, \\ D_a^+ & \text{with probability } 1/2. \end{cases} \tag{82}$$

Therefore, the sum of $f_{D_a^+}(t)$ and $f_{D_a^-}(t)$ is twice the pdf of $\chi^2(1, a^2)$, and the identity (80) readily follows from (27) and (74). ■

Remark 2. *Suppose that $a > 0$ is a real positive number, and that the functions $h_a^-, h_a^+ : \mathbb{R} \to \mathbb{C}$ are defined by (74), (80), respectively. Combining (74) with (80) and carrying out straightforward manipulations, we observe that*

$$\lim_{x\to\infty} h_a^-(x) \cdot \sqrt{x} = 0 \tag{83}$$

and that

$$\lim_{x\to\infty} h_a^+(x) \cdot \sqrt{x} = (1+i) \cdot e^{-a^2/2}. \tag{84}$$

The remainder of this section is devoted to generalizing Theorems 7, 8 to the multidimensional case.

Definition 8 (the random variable $D_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$)**.** *Suppose that $d \ge L \ge 3$ are positive integers and that $\boldsymbol{a} \in \mathbb{R}^d$ is an d-dimensional vector all of whose coordinates are positive. Suppose also that $\boldsymbol{\sigma}$ is a word of symbols $+, -$ of length L (see (3) in Section 2.1), and that $Q_{\boldsymbol{\sigma}}^d$ is a subset of $\mathbb{R}^d$ defined by (6). We define the random variable $D_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$ to be the square of the distance from $\boldsymbol{a}$ to the standard normal d-dimensional random vector conditioned on $Q_{\boldsymbol{\sigma}}^d$ (see Definition 4 in Section 2.3). In other words,*

$$D_{\boldsymbol{a}}^{\boldsymbol{\sigma}} = \|\boldsymbol{x}_d^{\boldsymbol{\sigma}} - \boldsymbol{a}\|^2 = \sum_{j=1}^d \left(x_d^{\boldsymbol{\sigma}}(j) - a(j) \right)^2, \tag{85}$$

where $\boldsymbol{x}_d^{\boldsymbol{\sigma}} \sim N(0_d, I_d) \mid Q_{\boldsymbol{\sigma}}^d$.

The following theorem provides the pdf of $D_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$.

Theorem 9. *Suppose that $D_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$ is a random variable as in Definition 8. We define the function $h_{\boldsymbol{a}}^{\boldsymbol{\sigma}} : \mathbb{R} \to \mathbb{C}$ by the formula*

$$h_{\boldsymbol{a}}^{\boldsymbol{\sigma}}(x) =$$
$$\left(\prod_{j=1}^{L} h_{a(j)}^{\sigma_j}(x) \right) \cdot \exp\left[\frac{ix}{1-2ix} \cdot \sum_{j=L+1}^{d} a(j)^2 \right] \cdot \left(\frac{1}{\sqrt{1-2ix}} \right)^{d-L}, \tag{86}$$

where h_a^- and h_a^+ are given respectively by (74), (80). Then the pdf $f_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$ of $D_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$ is given by the formula

$$f_{\boldsymbol{a}}^{\boldsymbol{\sigma}}(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-ixt} \cdot h_{\boldsymbol{a}}^{\boldsymbol{\sigma}}(x)\, dx. \tag{87}$$

Proof. We define the independent random variables $D_1, \dots, D_d$ by the formula

$$D_j = \begin{cases} D_{a(j)}^{\sigma_j} & 1 \le j \le L, \\ |N(0,1) - a(j)|^2 & L < j \le d, \end{cases} \tag{88}$$

where D_a^-, D_a^+ are defined respectively by (72), (78). Then due to (85)

$$D_{\boldsymbol{a}}^{\boldsymbol{\sigma}} = D_1 + \dots + D_d. \tag{89}$$

We denote the pdf of D_j for $j = 1, \dots, d$ by f_{D_j}. Also, we denote by $h_j : \mathbb{R} \to \mathbb{C}$ the Fourier transform of f_{D_j} (see (15) in Section 2.2). Due to independence of $D_1, \dots, D_d$ in (89), the pdf $f_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$ of $D_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$ is given by the convolution of f_{D_1} through f_{D_d}, i.e. for all real t,

$$f_{\boldsymbol{a}}^{\boldsymbol{\sigma}}(t) = (f_{D_1} * \dots * f_{D_d})(t). \tag{90}$$

Therefore, the Fourier transform of $f_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$ is given by the product of h_1 through h_d, i.e. for all real x,

$$\int_{-\infty}^{\infty} f_{\boldsymbol{a}}^{\boldsymbol{\sigma}}(t) \cdot e^{ixt}\, dt = \prod_{j=1}^{d} h_j(x). \tag{91}$$

For $j = 1, \dots, L$ and all real x,

$$h_j(x) = \int_{-\infty}^{\infty} f_{D_j}(t) \cdot e^{ixt}\, dt = h_{a(j)}^{\sigma_j}(x), \tag{92}$$

due to Theorems 7, 8. For $j = L+1, \dots, d$ and all real x,

$$h_j(x) = \int_{-\infty}^{\infty} f_{D_j}(t) \cdot e^{ixt}\, dt = \exp\left[\frac{ixa(j)^2}{1-2ix} \right] \cdot \frac{1}{\sqrt{1-2ix}}, \tag{93}$$

due to (27). We combine (87), (91), (92) and (93) to conclude that the Fourier transform of $f_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$ is given by the function $h_{\boldsymbol{a}}^{\boldsymbol{\sigma}} : \mathbb{R} \to \mathbb{C}$, defined via (87). Next, we observe that since $d \ge 3$, the function $h_{\boldsymbol{a}}^{\boldsymbol{\sigma}}(x)$ decays at infinity at least as fast as $x^{-3/2}$, due to (83), (84) in Remark 2 and identity (93). Since $h_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$ is also obviously continuous, it follows that $h_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$ belongs to $L^1(\mathbb{R})$, and the formula (87) is implied by the Fourier inversion formula (16) in Section 2.2. ■

Corollary 1. *The cdf* $F_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$ *of the random variable* $D_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$ *(see Definition 8) is given for all positive* $x > 0$ *by the formula*

$$F_{\boldsymbol{a}}^{\boldsymbol{\sigma}}(x) = \int_0^x f_{\boldsymbol{a}}^{\boldsymbol{\sigma}}(t)\, dt, \tag{94}$$

where $f_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$ *is the pdf of* $D_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$*, defined via* (87)*.*

4 The Randomized Approximate Nearest Neighbors algorithm (RANN)

In this section, we describe the Nearest Neighbor Problem and present a fast randomized algorithm for its solution.

4.1 The Nearest Neighbor Problem

Suppose that d and $k < N$ are positive integers and suppose that

$$B = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\} \subseteq \mathbb{R}^d \tag{95}$$

is a collection of N points in $\mathbb{R}^d$. We are interested in finding the k nearest neighbors of each point $\boldsymbol{x}_i$.

For each $\boldsymbol{x}_i$, one can compute in a straightforward manner the distances to the rest of the points and thus find the nearest neighbors. However, the total cost of the evaluation of the distances alone is $O(d \cdot N^2)$, which makes this naive approach prohibitively expensive when N is large. We propose a faster approximate algorithm for the solution of this problem.

4.2 Informal description of the algorithm

4.2.1 Initial selection

The key idea of our algorithm is the following simple (and well known) observation. Suppose that for each $\boldsymbol{x}_i$ we have found a small subset V_i of B such that a point inside V_i is more likely to be among the k nearest neighbors of $\boldsymbol{x}_i$ than a point outside V_i. Then it is reasonable to look for the nearest neighbors of each $\boldsymbol{x}_i$ only inside V_i and not among all the points. The nearest neighbors of $\boldsymbol{x}_i$ in V_i, which can be found by direct scanning, are referred to as its "suspected approximate nearest neighbors", or "suspects", as opposed to the true nearest neighbors $\left\{\boldsymbol{x}_{t(i,j)}\right\}$.

Of course, many of the k true nearest neighbors of $\boldsymbol{x}_i$ might not be among its suspects. However, one can re-select V_i to obtain another list of k suspects of $\boldsymbol{x}_i$. The initial guess is improved by taking the “best” k points out of the two lists. This scheme is iterated to successively improve the list of suspects of each $\boldsymbol{x}_i$.

The performance of the resulting iterative randomized algorithm admits the following crude analysis. Suppose that the size of V_i is $\alpha \cdot N$, with $\alpha \ll 1$. Suppose also that the number of the true nearest neighbors of $\boldsymbol{x}_i$ inside V_i is roughly $\beta \cdot k$, with $\alpha < \beta < 1$. If the choice of V_i is fairly random, then order $O(1/\beta)$ iterations of the algorithm are required to find most of the true nearest neighbors of each $\boldsymbol{x}_i$. Temporarily neglecting the cost of the construction of V_i, this results in $O\left((\alpha/\beta) \cdot d \cdot N^2\right)$ operations instead of $O\left(d \cdot N^2\right)$ operations for the naive algorithm. If $\alpha \ll \beta$, the improvement can be substantial.

Our construction of V_i's is based on geometric considerations. First, we shift all of the points to place their center of mass at the origin and apply a random orthogonal linear transformation on the resulting collection. (Later on, we will divide our sets according to the median - see Section 6.1. Here, for simplicity of presentation, as well as applications in the Gaussian case, we divide using the center of mass.) Then, we divide all the points in B into two disjoint sets

$$\begin{aligned} B_- &= \{\boldsymbol{x} \in B : x(1) < 0\}, \\ B_+ &= \{\boldsymbol{x} \in B : x(1) \geq 0\}. \end{aligned} \tag{96}$$

In other words, B_- consists of those points whose first coordinate is negative, and B_+ consists of those points whose first coordinate is non-negative. Next, we split B_+ into two disjoint sets B_{+-} and B_{++} by the same principle, but using the second coordinate, i.e.

$$\begin{aligned} B_{+-} &= \{\boldsymbol{x} \in B_+ : x(2) < 0\}, \\ B_{++} &= \{\boldsymbol{x} \in B_+ : x(2) \geq 0\}. \end{aligned} \tag{97}$$

We construct B_{--} and B_{-+} in a similar fashion via splitting B_- by the second coordinate of its points, i.e.

$$\begin{aligned} B_{--} &= \{\boldsymbol{x} \in B_- : x(2) < 0\}, \\ B_{-+} &= \{\boldsymbol{x} \in B_- : x(2) \geq 0\}. \end{aligned} \tag{98}$$

Then we repeat the subdivision by splitting each of the four boxes into two by using the third coordinate, and so on. We proceed until we end up with a collection of 2^L boxes $\{B_{\boldsymbol{\sigma}}\}$, containing k points on average. In other words, L is a positive integer defined via the inequality

$$k \cdot 2^L \leq N < k \cdot 2^{L+1}. \tag{99}$$

The box index $\boldsymbol{\sigma}$ is a word of symbols $+, -$ of length L, as in (3) in Section 2.1. We easily observe, that $B_{\boldsymbol{\sigma}}$ consists of those points $\boldsymbol{x}$ in B, the signs of whose first L coordinates coincide with the corresponding symbols of the word $\boldsymbol{\sigma}$. In other words,

$$B_{\boldsymbol{\sigma}} = \{\boldsymbol{x} \in B \ : \ \operatorname{sgn}(x(l)) = \sigma_l, \ \ l = 1, \ldots, L\} = B \cap Q^d_{\boldsymbol{\sigma}}, \tag{100}$$

due to (6) in Section 2.1. Obviously, the sets $\left\{B_{\boldsymbol{\mu}}\right\}$ constitute a complete binary tree of length L, whose nodes are indexed by words $\boldsymbol{\mu}$ of symbols $+, -$ of length up to L. The set B is at the root of this tree, the sets B_- and B_+ are at the second level, and so on. The 2^L boxes $B_{\boldsymbol{\sigma}}$, defined via (100), are at the Lth (and last) level of the tree.

The construction is illustrated in Figure 1(a). Here, the parameters are $k = 5$ and $d = L = 2$, and the number of points is $N = 5 \cdot 2^2 = 20$. Thus we have 4 boxes, each containing 5 points on average. More specifically, the boxes B_{++}, B_{+-}, B_{--} and B_{-+} contain 8, 2, 7 and 3 points, respectively.

The notion of degree of contact (4) extends to the collection $\{B_{\boldsymbol{\sigma}}\}$ of boxes. Suppose that $\boldsymbol{x}_i$ is in $B_{\boldsymbol{\sigma}}$. Obviously, the higher degree of contact of two boxes $B_{\boldsymbol{\sigma}}$ and $B_{\boldsymbol{\mu}}$ is, the less likely a point of $B_{\boldsymbol{\mu}}$ will be among the k nearest neighbors of $\boldsymbol{x}_i$. Motivated by this observation, we define the set V_i as

$$V_i = \left\{ \boldsymbol{x} \in B_{\boldsymbol{\mu}} \ : \ Con(\boldsymbol{\sigma}, \boldsymbol{\mu}) \leq 1 \right\} = \bigcup_{j=0}^{L} B_{\boldsymbol{\sigma}^j}, \tag{101}$$

due to Definition 1 in Section 2.1. In other words, V_i is the union of the box $B_{\boldsymbol{\sigma}}$ containing $\boldsymbol{x}_i$ and L boxes whose degree of contact with $B_{\boldsymbol{\sigma}}$ is one. Thus for each $i = 1, \ldots, N$, the set V_i contains about $k \cdot (L+1)$ points on average. For example, in Figure 1(a) for every point $\boldsymbol{x}_i$ in B_{++} (upper right box), the set V_i is the union of B_{++}, B_{-+} and B_{+-}, containing 12 points. On the other hand, for every point $\boldsymbol{x}_i$ in B_{+-} (lower right box), the set V_i is the union of B_{+-}, B_{++} and B_{--}, containing 18 points.

The choice of suspects is illustrated in Figure 1(b). The division into boxes is the same as in Figure 1(a). The point $\boldsymbol{x}_i$ is the uppermost point in B_{--}. Its 5 true nearest neighbors are marked with squares. The 5 suspects of $\boldsymbol{x}_i$ are connected to it by lines. One of the true nearest neighbors is not among them, since it belongs to B_{++}, and the degree of contact between B_{--} and B_{++} is two.

In Section 6.1, an alternative (though very similar) construction of the boxes $\{B_{\boldsymbol{\sigma}}\}$ is proposed. This construction is also mentioned in Section 1.

4.2.2 "Supercharging"

In Section 4.2.1, we have described an iterative scheme for the selection of suspects (suspected approximate nearest neighbors) for each of the points $\boldsymbol{x}_i$ in B. Suppose now that after T iterations of this scheme, the list $\boldsymbol{x}_{s(i,1)}, \ldots, \boldsymbol{x}_{s(1,k)}$ of k suspects of each point $\boldsymbol{x}_i$ has been generated. This list can be improved by a procedure we call supercharging.

The idea of supercharging is based on the following observation. A true nearest neighbor of $\boldsymbol{x}_i$, missed by the scheme described above, might be among the suspects of one of $\boldsymbol{x}_{s(i,1)}, \ldots, \boldsymbol{x}_{s(1,k)}$. This leads to the following obvious procedure.

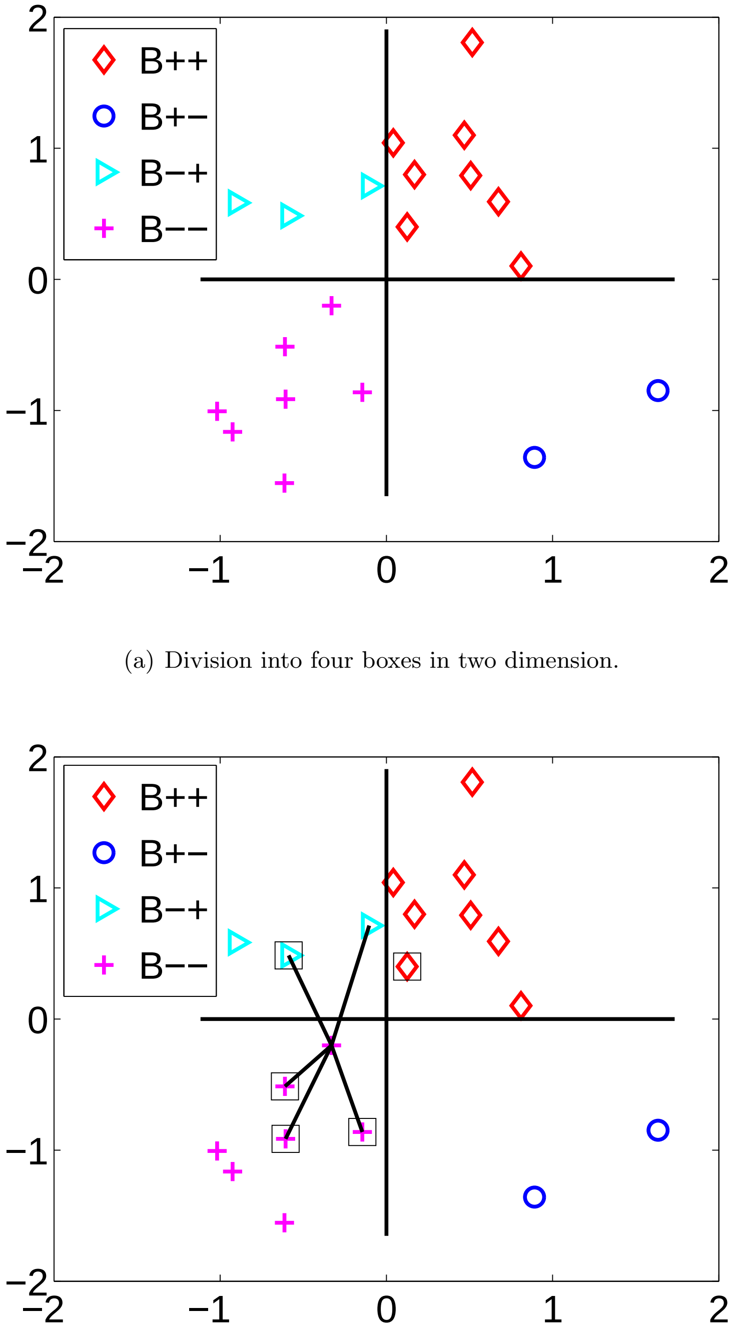
For each $\boldsymbol{x}_i$, we denote by A_i the list of suspects of all $\boldsymbol{x}_{s(i,1)}, \ldots, \boldsymbol{x}_{s(i,k)}$. A_i contains k^2 points, with possible repetitions. We compute the square of the distances from $\boldsymbol{x}_i$ to each point in A_i and find the k nearest neighbors $\boldsymbol{x}_{t(i,1,A_i)}, \ldots, \boldsymbol{x}_{t(i,k,A_i)}$ of $\boldsymbol{x}_i$ in A_i. Then we declare the (updated) suspects of $\boldsymbol{x}_i$ to be the best k points out of the two lists $\left\{\boldsymbol{x}_{s(i,j)}\right\}_{j=1}^{k}$ and $\left\{\boldsymbol{x}_{t(i,j,A_i)}\right\}_{j=1}^{k}$.

In other words, supercharging is a depth one search on the graph, whose vertices are the points $\{\boldsymbol{x}_i\}$ and whose $N \times k$ adjacency matrix is the suspects' indices $\{s(i,j)\}$, with $i = 1, \ldots, N$ and $j = 1, \ldots, k$.

4.2.3 Overview

We conclude this section with a list of the principal steps of the algorithm. Given the collection $\{\boldsymbol{x}_i\}_{i=1}^{N}$ of points in $\mathbb{R}^d$, we perform the following operations.

1. Subtract from each $\boldsymbol{x}_i$ the center of mass of the collection.



(a) Division into four boxes in two dimension.

(b) Suspects vs. true nearest neighbors.

Figure 1: Illustration of the algorithm in two dimension.

2. Choose a random orthogonal linear transformation Θ and set $\boldsymbol{x}_i = \Theta(\boldsymbol{x}_i)$ for all $i = 1, \ldots, N$.

3. Construct 2^L boxes $\{B_{\boldsymbol{\sigma}}\}$ as described in Section 4.2.1 (see (100)).

4. For each $\boldsymbol{x}_i$ define the set V_i via (101).

5. Update the suspects $\boldsymbol{x}_{s(i,1)}, \ldots, \boldsymbol{x}_{s(i,k)}$ of $\boldsymbol{x}_i$ by using its true nearest neighbors in V_i.

6. Steps 2-5 are repeated T times.

7. For each $\boldsymbol{x}_i$, perform supercharging.

4.2.4 Query for a new point

Suppose that we are given a new point $\boldsymbol{y} \in \mathbb{R}^d$, and we need to find its k nearest neighbors in $B = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$. In this section, we describe a rapid procedure to find k approximate nearest neighbors of $\boldsymbol{y}$. This procedure uses the following information, available on the jth iteration of the algorithm, for $j = 1, \ldots, T$:

1. The orthogonal linear transformation $\Theta^{(j)}$, generated on the jth iteration of the algorithm (Step 2 in Section 4.2.3).

2. The collection of boxes $\left\{B_{\boldsymbol{\sigma}}^{(j)}\right\}$, generated on the jth iteration of the algorithm (Step 3 in Section 4.2.3).

3. For each point $\boldsymbol{x}_i$, the list of its k nearest neighbors in the union $V_i^{(j)}$ of "close" boxes (Step 4 in Section 4.2.3).

To find k approximate nearest neighbors of the new point $\boldsymbol{y}$ among the points of B, we perform the following operations. First, we apply $\Theta^{(1)}$ on $\boldsymbol{y}$, where $\Theta^{(1)}$ is the orthogonal linear transformation of the first iteration of the algorithm. The resulting vector is denoted by $\boldsymbol{y}^{(1)}$, in other words,

$$\boldsymbol{y}^{(1)} = \Theta^{(1)}(\boldsymbol{y}). \tag{102}$$

Next, in the collection of boxes $\left\{B_{\boldsymbol{\sigma}}^{(1)}\right\}$, generated on the first iteration of the algorithm, we find the box $B_{\boldsymbol{\sigma}(1)}^{(1)}$ that has degree of contact zero with $\boldsymbol{y}^{(1)}$. In other words, for each $l = 1, \ldots, L$, the lth symbol of $\boldsymbol{\sigma}(1)$ is the sign of lth coordinate of $\boldsymbol{y}^{(1)}$, i.e.

$$\boldsymbol{\sigma}(1) = \left(\operatorname{sgn}\left(y^{(1)}(1)\right), \ldots, \operatorname{sgn}\left(y^{(1)}(L)\right)\right). \tag{103}$$

Note, that if $\boldsymbol{y}$ had belonged to B in the first place, then on the first iteration of the algorithm $\boldsymbol{y}^{(1)}$ would have belonged to $B_{\boldsymbol{\sigma}(1)}^{(1)}$.

The box $B_{\boldsymbol{\sigma}(1)}^{(1)}$ has roughly k points. Each point $\boldsymbol{x}_i$ in $B_{\boldsymbol{\sigma}(1)}^{(1)}$ has a list of its k nearest neighbors in the set $V_i^{(1)}$, where, similar to (101), $V_i^{(1)}$ is the union of the boxes having degree of contact zero or one with $B_{\boldsymbol{\sigma}(1)}^{(1)}$. We denote by $A^{(1)}$ the union of the nearest neighbors of each $\boldsymbol{x}_i \in B^{(1)}_{\boldsymbol{\sigma}(1)}$ among $V_i^{(1)}$. Note that the set $A^{(1)}$ has roughly k^2 points, with possible repetitions.

We construct the set $A^{(2)}$ in a similar manner, by using the data of the second iteration of the algorithm. We apply the orthogonal transformation $\Theta^{(2)}$ of the second iteration on $\boldsymbol{y}^{(1)}$ to obtain $\boldsymbol{y}^{(2)}$, i.e.

$$\boldsymbol{y}^{(2)} = \Theta^{(2)}\left(\boldsymbol{y}^{(1)}\right) = \Theta^{(2)}\left(\Theta^{(1)}\left(\boldsymbol{y}\right)\right), \tag{104}$$

due to (102). In the boxes $\left\{B^{(2)}_{\boldsymbol{\sigma}}\right\}$ of the second iteration, we find the box $B^{(2)}_{\boldsymbol{\sigma}(2)}$, having degree of contact zero with $\boldsymbol{y}^{(2)}$. Each $\boldsymbol{x}_i$ in $B^{(2)}_{\boldsymbol{\sigma}(2)}$ has a list of k nearest neighbors of $\boldsymbol{x}_i$ in $V_i^{(2)}$, and we denote their union by $A^{(2)}$. Similar to $A^{(1)}$, the set $A^{(2)}$ contains roughly k^2 points.

We repeat this procedure to constructs the sets $A^{(j)}$ for $j = 3, 4, \dots, T$, where T is the number of the iterations of the algorithm. Each $A^{(j)}$ contains roughly k^2 points.

Finally, we define the set A to be the union of all the sets $A^{(j)}$, in other words,

$$A = \bigcup_{j=1}^{T} A^{(j)}. \tag{105}$$

The set A contains roughly $T \cdot k^2$ points. The k nearest neighbors of $\boldsymbol{y}$ inside A are declared to be the approximate nearest neighbors of $\boldsymbol{y}$ inside B. We note that to construct A we need to store the corresponding data on each iteration of the algorithm.

4.3 Detailed description of the algorithm

Input

A collection $B = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ of points in $\mathbb{R}^d$, the number $0 < k < N$ of required nearest neighbors, the number $T > 0$ of iterations.

Output

For each point $\boldsymbol{x}_i$, return the list $\left\{\boldsymbol{x}_{s(i,1)}, \dots \boldsymbol{x}_{s(i,k)}\right\}$ of its k suspects and the square of the distances to them, $d_{i,j} = \|\boldsymbol{x}_i - \boldsymbol{x}_{s(i,j)}\|^2$.

4.3.1 Initialization

Step 1.

Shift all the points by the center of mass of the collection.

Comment. Now the points are centered about the origin.

4.3.2 A single iteration of the algorithm

The Steps 2, 3, 4 below are repeated T times.

Step 2.

Choose a pseudorandom orthogonal linear transformation Θ and apply it to every point $\boldsymbol{x}_i$, with $i = 1, \ldots, N$, as described in Section 2.4. Note that the distances between the points are preserved.

Step 3.

Choose the number of subdivisions to be $L = \lfloor \log_2 (N/k) \rfloor$.

do $l = 1, \ldots, L$

do for all 2^{l-1} words $\boldsymbol{\mu} = \mu_1 \ldots \mu_{l-1}$ of symbols $+, -$

Split the box $B_{\boldsymbol{\mu}}$ into two boxes $B_{\boldsymbol{\mu}-}$ and $B_{\boldsymbol{\mu}+}$, such that the lth coordinate of any point in $B_{\boldsymbol{\mu}-}$ is negative and the lth coordinate of any point in $B_{\boldsymbol{\mu}+}$ is non-negative. In other words,

$$\begin{aligned} B_{\boldsymbol{\mu}-} &= \left\{ \boldsymbol{x} \in B_{\boldsymbol{\mu}} \ : \ x(l) < 0 \right\}, \\ B_{\boldsymbol{\mu}+} &= \left\{ \boldsymbol{x} \in B_{\boldsymbol{\mu}} \ : \ x(l) \geq 0 \right\}. \end{aligned} \tag{106}$$

Comment. See (96), (97), (98), (100) in Section 4.2.1).

Step 4.

do for all 2^L words $\boldsymbol{\sigma} = \sigma_1 \ldots \sigma_L$ of symbols $+, -$

A. Set $V_{\boldsymbol{\sigma}}$ to be the union of $B_{\boldsymbol{\sigma}}$ and the L boxes $B_{\boldsymbol{\mu}}$ having degree of contact one with $B_{\boldsymbol{\sigma}}$, as defined by (4). In other words,

$$V_{\boldsymbol{\sigma}} = \left\{ \boldsymbol{x} \in B_{\boldsymbol{\mu}} \ : \ Con\left(\boldsymbol{\sigma}, \boldsymbol{\mu}\right) \leq 1 \right\} = \bigcup_{j=0}^{L} B_{\boldsymbol{\sigma}^j}, \tag{107}$$

due to Definition 1. $V_{\boldsymbol{\sigma}}$ contains $k \cdot (L+1)$ points on average.

B. For each point $\boldsymbol{x}_i$ in $B_{\boldsymbol{\sigma}}$, find the list $\boldsymbol{x}_{t(i,1,V_{\boldsymbol{\sigma}})}, \ldots, \boldsymbol{x}_{t(i,k,V_{\boldsymbol{\sigma}})}$ of its k nearest neighbors in $V_{\boldsymbol{\sigma}}$.

Comment. A heap of size k is used to find $\left\{ \boldsymbol{x}_{t(i,j,V_{\boldsymbol{\sigma}})} \right\}_{j=1}^{k}$.

C. Update the suspects $\boldsymbol{x}_{s(i,1)}, \ldots, \boldsymbol{x}_{s(i,k)}$ of each $\boldsymbol{x}_i$ in $B_{\boldsymbol{\sigma}}$ by taking the best k points out of the two lists $\left\{ \boldsymbol{x}_{t(i,j,V_{\boldsymbol{\sigma}})} \right\}_{j=1}^{k}$ and $\left\{ \boldsymbol{x}_{s(i,j)} \right\}_{j=1}^{k}$.

Comment. The two lists are merged and sorted according to the distances to $\boldsymbol{x}_i$. Then the redundant indices are removed and the first k points are declared to be the updated suspects of $\boldsymbol{x}_i$.

4.3.3 Supercharging

Step 5 below is carried out once after T iterations of Steps 2, 3, 4.

Step 5.

do for all $i = 1, \dots, N$

A. Set A_i to be the list of suspects of $\boldsymbol{x}_{s(i,1)}, \dots, \boldsymbol{x}_{s(i,k)}$, i.e.

$$A_i = \left\{\boldsymbol{x}_{s(s(i,j),l)}\right\}_{j,l=1}^{k}. \tag{108}$$

A_i contains k^2 points with possible repetitions.

B. Compute the distances from $\boldsymbol{x}_i$ to each of the k^2 points in A_i.

C. Find the k nearest neighbors $\boldsymbol{x}_{t(i,1,A_i)}, \dots, \boldsymbol{x}_{t(i,k,A_i)}$ of $\boldsymbol{x}_i$ in A_i.

Comment. A heap of size k is used and the redundancies (i.e. second appearances of the same point) are removed.

D. Update the suspects $\boldsymbol{x}_{s(i,1)}, \dots, \boldsymbol{x}_{s(i,k)}$ of $\boldsymbol{x}_i$ by taking the best k points out of the two lists $\left\{\boldsymbol{x}_{t(i,j,A_i)}\right\}_{j=1}^{k}$ and $\left\{\boldsymbol{x}_{s(i,j)}\right\}_{j=1}^{k}$.

4.3.4 Query for a new point

Given a new point $\boldsymbol{y} \in \mathbb{R}^d$, we find its k approximate nearest neighbors in the set $B = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$. We assume that the algorithm has already found k suspects for each $\boldsymbol{x}_i$, and, moreover, the relevant data for each of T iterations have been stored.

1. Define $\boldsymbol{y}^{(0)} = \boldsymbol{y}$.

2. **do** for all $j = 1, \dots, T$

A. Define the vector $\boldsymbol{y}^{(j)}$ by the formula

$$\boldsymbol{y}^{(j)} = \Theta^{(j)}\left(y^{(j-1)}\right), \tag{109}$$

where $\Theta^{(j)}$ is the orthogonal transformation of Step 2 of the jth iteration (see also (102), (104)).

B. Define the word $\boldsymbol{\sigma}(j)$ of symbols $+, -$ of length L by the formula

$$\boldsymbol{\sigma}(j) = \left(\operatorname{sgn}\left(y^{(j)}(1)\right), \dots, \operatorname{sgn}\left(y^{(j)}(L)\right)\right), \tag{110}$$

similar to (103). Out of the collection $\left\{B^{(j)}_{\boldsymbol{\sigma}}\right\}$ generated on Step 3 of the jth iteration, the box $B^{(j)}_{\boldsymbol{\sigma}(j)}$ has degree of contact zero with $\boldsymbol{y}^{(j)}$.

Comment. Since the boxes $\left\{B^{(j)}_{\boldsymbol{\sigma}}\right\}$ are stored on the leaves of a binary tree of length L (see Section 4.2.1), the box $B^{(j)}_{\boldsymbol{\sigma}(j)}$ can be found by the binary search on this tree. This search method also works for the alternative construction of boxes, described in Section 6.1.

C. For each point $\boldsymbol{x}_i$ in $B^{(j)}_{\boldsymbol{\sigma}(j)}$, define $A^{(j)}_i$ to be the list of k the nearest neighbors of $\boldsymbol{x}_i$ in $V^{(j)}_{\boldsymbol{\sigma}(j)}$, where

$$V^{(j)}_{\boldsymbol{\sigma}(j)} = \left\{ \boldsymbol{x} \in B^{(j)}_{\boldsymbol{\mu}} \ : \ Con\left(\boldsymbol{\sigma}(j), \boldsymbol{\mu}\right) \leq 1 \right\}, \tag{111}$$

similar to (107). In other words,

$$A^{(j)}_i = \left\{ \boldsymbol{x}_{t(i,1,V)}, \ldots, \boldsymbol{x}_{t(i,k,V)} \right\}, \tag{112}$$

where $V = V^{(j)}_{\boldsymbol{\sigma}(j)}$. Then, define $A^{(j)}$ to be the union of all $A^{(j)}_i$, for $\boldsymbol{x}_i \in B^{(j)}_{\boldsymbol{\sigma}(j)}$. In other words,

$$A^{(j)} = \left\{ \boldsymbol{x} \in A^{(j)}_i \ : \ \boldsymbol{x}_i \in B^{(j)}_{\boldsymbol{\sigma}(j)} \right\}. \tag{113}$$

The set $A^{(j)}$ contains roughly k^2 points (see also Section 4.2.4).

3. Define the set A to be the union of all $A^{(j)}$. In other words, A is defined via (105). The set A contains roughly $T \cdot k^2$ points.

4. Compute the distances from $\boldsymbol{y}$ to each of the points $\boldsymbol{x}_i$ in A.

5. Find the k nearest neighbors of $\boldsymbol{y}$ in A (similar to C in Step 5, see Section 4.3.3). These are the approximate nearest neighbors of $\boldsymbol{y}$ in B.

4.4 Cost analysis

In this section, we analyze the cost of the algorithm in terms of number of operations. Also, we analyze the memory requirements of the algorithm. We recall that $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ is a collection of N points in $\mathbb{R}^d$ and $N \approx k \cdot 2^L$. We estimate the number of operations for each step described in the preceding Section 4.3.

The cost of Step 1.

- It takes $2 \cdot d \cdot N$ operations to centralize the points.

The cost of Step 2.

- In our implementation of the pseudorandom orthogonal transformation algorithm (see Section 2.4), the parameters in the formula (53) were chosen to be $M_1 = 1$, $M_2 = 6$. Therefore it takes $O(d \cdot (\log d + 7)) = O(d \cdot (\log d))$ operations to generate a random transformation Θ, and also $O(d \cdot (\log d))$ operations to apply it to each point $\boldsymbol{x}_i$ (see (62) in Section 2.4). Thus the total cost of this step is $O(N \cdot d \cdot (\log d))$ operations.

The cost of Step 3.

- The cost of splitting a single box $B_{\boldsymbol{\mu}}$ into two is order $O(|B_{\boldsymbol{\mu}}|)$ operations.
- There are 2^{l-1} boxes $B_{\boldsymbol{\mu}}$ for each $l = 1, \dots, L$, containing N points altogether, which results in $O(N)$ operations for each l.
- Hence the total cost is $O(L \cdot N)$.

The cost of Step 4.

- The cost of finding the k nearest neighbors of each $\boldsymbol{x}_i$ inside $V_{\boldsymbol{\sigma}}$ of size about $k \cdot (L+1)$ is $O(L \cdot k \cdot (\log k))$.
- The cost of updating the suspects of each $\boldsymbol{x}_i$ is $O(k \cdot (\log k))$.
- Hence the total cost of this step is $O(N \cdot L \cdot k \cdot (\log k))$.

The cost of Step 5.

- The cost of computing the distances to k^2 points for each $\boldsymbol{x}_i$ is $O(d \cdot k^2)$.
- The cost of finding the best k points out of this list is $O(k^2 \cdot (\log k))$.
- Hence the total cost of supercharging is $O\left(N \cdot k^2 \cdot (d + \log k)\right)$.

The total cost of the algorithm

We conclude that the total cost of the algorithm is

$$O\left(T \cdot N \cdot (d \cdot (\log d) + k \cdot (\log k) \cdot (\log N))\right) + O(N \cdot k^2 \cdot (d + \log k)), \tag{114}$$

where T is the number of iterations. We observe that for fixed dimension d and number of required nearest neighbors k, the cost is $O(T \cdot N \cdot \log N)$, as opposed to $O(N^2)$ of the naive approach. Also, the cost of supercharging is quadratic in the number of nearest neighbors for fixed dimension d and number of points N, which makes supercharging expensive relative to a single iteration of the principal part of the algorithm even for moderate k.

The cost of query for a new point

- The cost of computing $y^{(j)}$ for each $j = 1, \dots, T$ is order $O(d \cdot (\log d))$ operations (see the cost of Step 2).
- The cost of finding $\boldsymbol{\sigma}(j)$, defined via (110), is $O(L) = O(\log_2(N/k))$ operations (since this is the binary search in a binary tree of length L).
- The cost of the construction of $A^{(j)}$, defined via (113), is order k^2 operations.

- Thus the total cost of the construction of A, defined via (105), is order
$$O\left(T \cdot \left(d \cdot (\log d) + \log(N/k) + k^2\right)\right). \tag{115}$$
- The cost of the evaluation of the distances from $\boldsymbol{y}$ to each point $\boldsymbol{x}_i$ in A is order $O(T \cdot d \cdot k^2)$ operations, since A contains roughly $T \cdot k^2$ points in $\mathbb{R}^d$.
- The cost of finding k nearest neighbors of $\boldsymbol{y}$ among the points of A is order $O(T \cdot k^2 \cdot (\log k))$ operations.
- Thus the total cost of query for a new point $\boldsymbol{y}$ is order
$$O\left(T \cdot \left(d \cdot (\log d) + \log(N/k) + k^2 \cdot (d + \log k)\right)\right). \tag{116}$$
We observe that this cost grows linearly in the number of iterations T, grows as $d \cdot (\log d)$ in the dimension d, grows as $\log N$ in the number of points N, and grows as $k^2 \cdot (\log k)$ in the number of requested nearest neighbors k.

Memory requirements

We must distinguish between two cases. In the first case, given N points $\{\boldsymbol{x}_i\}$ in $\mathbb{R}^d$, we are interested in finding k nearest neighbors for each $\boldsymbol{x}_i$ only. In other words, no query for a new point will ever be requested. Then, the memory requirements of the algorithm are order $O(N \cdot (d + k))$, since:

- the memory required to store N points in $\mathbb{R}^d$ is of the order $O(N \cdot d)$;
- the memory required to store the indices of k nearest neighbors of each of N points is of the order $O(N \cdot k)$;
- the memory required to store $2^L \approx N/k$ boxes $\{B_{\boldsymbol{\sigma}}\}$ and the corresponding binary tree, constructed on Step 3 of Section 4.3, is of the order $O(N)$.

In other words, in this case the memory requirements are minimal, in the sense that most of the memory is spent on the storage of input and output of the algorithm only.

In the second case, we know in advance that queries for new points will be requested. Differently put, the environment is dynamic, i.e. not all data are known at the time of the first invocation of the algorithm. To perform the query for a new point $\boldsymbol{y} \in \mathbb{R}^d$ (see Section 4.2.4, 4.3.4), we need to store additional data on each iteration of the algorithm. The memory requirements are then as follows:

- The memory required to store the orthogonal transformation on Step 2 in Section 4.3 is $O(d)$.
- The memory required to store the boxes $\{B_{\boldsymbol{\sigma}}\}$ and the corresponding binary tree on Step 3 in Section 4.3 is $O(N)$.
- The memory required to store k neighbors of each point $\boldsymbol{x}_i$ is $O(N \cdot k)$ (see (112)).
- The memory required to store $A^{(j)}$, as defined by (113), is $O(k^2)$, for each $j = 1, \ldots, T$.

Thus, when queries for new points are allowed, the total memory requirements are of the order

$$O\left(N \cdot (d+k) + T \cdot (d + N \cdot k)\right). \tag{117}$$

4.5 Performance analysis

In this section, we analyze the performance of the Randomized Approximate Nearest Neighbor algorithm, described in Sections 4.2, 4.3.

We recall that for each point $\boldsymbol{x}_i$ out of a collection of N points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$ in $\mathbb{R}^d$, the algorithm approximates the k true nearest neighbors $\left\{\boldsymbol{x}_{t(i,j)}\right\}_{j=1}^{k}$ of $\boldsymbol{x}_i$ by k suspects $\left\{\boldsymbol{x}_{s(i,j)}\right\}_{j=1}^{k}$ (see Sections 4.1, 4.2.3). In order to analyze the quality of this approximation, we introduce a number of statistical quantities. First, we define the average square of the distance from $\boldsymbol{x}_i$ to its k true nearest neighbors by the formula

$$D_i^{true} = \frac{1}{k} \sum_{j=1}^{k} \|\boldsymbol{x}_i - \boldsymbol{x}_{t(i,j)}\|^2. \tag{118}$$

Next, we define the average square of the distance from $\boldsymbol{x}_i$ to its k suspects by the formula

$$D_i^{susp} = \frac{1}{k} \sum_{j=1}^{k} \|\boldsymbol{x}_i - \boldsymbol{x}_{s(i,j)}\|^2. \tag{119}$$

Finally, we define the proportion of the true nearest neighbors of $\boldsymbol{x}_i$ among its suspects by the formula

$$\text{prop}_i = \frac{1}{k} \left| \left\{\boldsymbol{x}_{t(i,j)}\right\}_{j=1}^{k} \cap \left\{\boldsymbol{x}_{s(i,j)}\right\}_{j=1}^{k} \right|. \tag{120}$$

To be able to analyze the quantities (118), (119), (120), we need to make some assumption on the distribution of the points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$. The most natural candidate is the standard normal distribution. To be more specific, we consider the collection of N independent standard normal d-dimensional random vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$ (see Section 2.3), where the number of points is given by the formula

$$N = k \cdot 2^L \tag{121}$$

for some positive integer $L > 0$, in agreement with Section 4.3.2.

We analyze a single iteration of the algorithm, as described by Steps 3, 4 of Section 4.3.2. Since the standard normal distribution is radially symmetric, the orthogonal transformation in Step 2 of Section 4.3.2 does not affect the distribution of the points. The effects of supercharging (Sections 4.2.2, 4.3.3) are not included in the analysis below.

4.5.1 Average distance to true nearest neighbors

In this section, we study the expectation of D_i^{true}, defined via (118). Obviously, it does not depend on i. Therefore, it suffices to compute $\mathbb{E}\left[D_i^{true}\right]$ for $i = N$ only. The following theorem provides an analytical formula for $\mathbb{E}\left[D_N^{true}\right]$.

Theorem 10. *Suppose that $d, k, N > 0$ are positive integers. Suppose further that D_N^{true} is defined by* (118)*. Then its expectation is given by the formula*

$$\mathbb{E}\left[D_N^{true}\right] = \frac{1}{k} \int_{\lambda=0}^{\infty} \left(\sum_{i=1}^{k} \int_0^1 F^{-1}_{\chi^2(d,\lambda)}(t) \cdot f_{B(i,N-i)}(t)\, dt \right) \cdot f_{\chi^2_d}(\lambda)\, d\lambda, \tag{122}$$

where the functions $f_{\chi^2_d}, f_{B(i,N-i)}$ are defined respectively by (23), (28) *in Section 2.3, and $F^{-1}_{\chi^2(d,\lambda)}$ is the inverse of the cdf of $\chi^2(d, \lambda)$ (see* (25) *in Section 2.3).*

Proof. We fix a vector $\boldsymbol{a} \in \mathbb{R}^d$ and consider $N-1$ i.i.d. standard normal d-dimensional random vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N-1}$. For $i = 1, \ldots, N-1$, we define the random variables $Y_i^{\boldsymbol{a}}$ by the formula

$$Y_i^{\boldsymbol{a}} = \|\boldsymbol{x}_i - \boldsymbol{a}\|^2. \tag{123}$$

The random variables $Y_i^{\boldsymbol{a}}$ are i.i.d., and

$$Y_i^{\boldsymbol{a}} \sim \chi^2(d, \lambda), \tag{124}$$

with $\lambda = \|\boldsymbol{a}\|^2$, due to (25) in Section 2.3. Due to Definition 3 in Section 2.3, the order statistics $Y_{(1)}^{\boldsymbol{a}}, \ldots, Y_{(k)}^{\boldsymbol{a}}$ are the squares of the distances to the k nearest neighbors of $\boldsymbol{a}$ among $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N-1}$. (Needless to say, the distribution of $Y_{(i)}^{\boldsymbol{a}}$ differs from that of $Y_i^{\boldsymbol{a}}$.) Therefore, the conditional expectation of D_N^{true} given that $\boldsymbol{x}_N = \boldsymbol{a}$ is provided by the formula

$$\mathbb{E}\left[D_N^{true} \mid \boldsymbol{x}_N = \boldsymbol{a}\right] = \frac{1}{k} \sum_{i=1}^{k} \mathbb{E}\left[Y_{(i)}^{\boldsymbol{a}}\right], \tag{125}$$

due to (118). To evaluate the expectation of D_N^{true}, we integrate (125) with respect to the pdf of $\boldsymbol{x}_N$ to obtain

$$\mathbb{E}\left[D_N^{true}\right] = \int_{\mathbb{R}^d} \mathbb{E}\left[D_N^{true} \mid \boldsymbol{x}_N = \boldsymbol{a}\right] \, d\mathbb{P}\, \{\boldsymbol{x}_N = \boldsymbol{a}\}. \tag{126}$$

To compute the integrand in (126), we need to evaluate each summand in (125). Due to Theorem 5 in Section 3,

$$\mathbb{E}\left[Y_{(i)}^{\boldsymbol{a}}\right] = \int_0^1 F^{-1}_{\chi^2(d,\lambda)}(t) \cdot f_{B(i,N-i)}(t)\, dt, \tag{127}$$

for all $i = 1, \ldots, N-1$. We observe that the right-hand side of (127) depends on $\boldsymbol{a}$ only through the square of its norm $\lambda = \|\boldsymbol{a}\|^2$. We combine this observation with (23), (125), (126), and (127) to establish (122). ■

The following theorem provides an approximation to (122) by a one-dimensional integral.

Theorem 11. *Suppose that $d, k, N > 0$ are positive integers, and that $2k < N$. Suppose further that D_N^{true} is defined by (118). We define D_{appr}^{true} by the formula*

$$D_{appr}^{true} = \frac{1}{k} \int_{\lambda=0}^{\infty} \left(\sum_{i=1}^{k} F_{\chi^2(d,\lambda)}^{-1} \left(\frac{i}{N} \right) \right) \cdot f_{\chi_d^2}(\lambda) \, d\lambda, \tag{128}$$

where the function $f_{\chi_d^2}$ is defined by (23) in Section 2.3, and $F_{\chi^2(d,\lambda)}^{-1}$ is the inverse of the cdf of $\chi^2(d, \lambda)$ (see (25) in Section 2.3). Then

$$\mathbb{E}\left[D_N^{true}\right] = D_{appr}^{true} + O\left(d\right), \quad d \to \infty, \tag{129}$$

and

$$\mathbb{E}\left[D_N^{true}\right] = D_{appr}^{true} + O\left(\frac{1}{\sqrt{k}}\right), \quad k, N \to \infty, \quad \frac{k}{N} = const. \tag{130}$$

In other words, the identity (129) holds, if we fix N, k and let $d \to \infty$, and the identity (130) holds, if we fix d, k/N and let $N \to \infty$.

Proof. We consider the integral (127) and observe that its integrand contains the pdf $f_{B(i,N-i)}$ of the beta distribution, with $1 \leq i \leq k$. Due to (29) in Section 2.3,

$$\begin{aligned} \mu_i &= \mathbb{E}\left[B(i, N-i)\right] = \frac{i}{N}, \\ \sigma_i^2 &= \mathrm{Var}\left[B(i, N-i)\right] = \frac{i}{N^2} \cdot \frac{N-i}{N+1}. \end{aligned} \tag{131}$$

We expand $F_{\chi^2(d,\lambda)}^{-1}(t)$ in (127) into a Taylor series about μ_i, namely, for all $0 < t < 1$,

$$F_{\chi^2(d,\lambda)}^{-1}(t) = F_{\chi^2(d,\lambda)}^{-1}(\mu_i) + (t - \mu_i) \cdot \frac{dF_{\chi^2(d,\lambda)}^{-1}}{dt}(\mu_i) + O\left((t-\mu_i)^2\right). \tag{132}$$

We substitute (132) into (127) and use (131) to obtain the formula

$$\mathbb{E}\left[Y_{(i)}^{\boldsymbol{a}}\right] = F_{\chi^2(d,\lambda)}^{-1}\left(\frac{i}{N}\right) + \int_0^1 O\left((t-\mu_i)^2\right) \cdot f_{B(i,N-i)}(t) \, dt. \tag{133}$$

On the other hand, the combination of (26), (124), the assumption that $2k < N$ and Theorem 1 in Section 2.3 implies that

$$\mathbb{E}\left[Y_{(i)}^{\boldsymbol{a}}\right] \leq 2 \cdot \mathbb{E}\left[\chi^2(d,\lambda)\right] = 2\left(d + \lambda\right), \tag{134}$$

for $i = 1, \ldots, k$. Therefore, due to (24) in Section 2.3,

$$\int_{\lambda=0}^{\infty} \mathbb{E}\left[Y_{(i)}^{\boldsymbol{a}}\right] \cdot f_{\chi_d^2}(\lambda) \, d\lambda = O(d). \tag{135}$$

We combine (135) with (122), (125), (126) to conclude that

$$\mathbb{E}\left[D_{appr}^{true}\right] = O(d). \tag{136}$$

The identity (129) then follows from the combination of (133) and (136). The identity (130) is a direct consequence of Theorem 6 in Section 3. ∎

Remark 3. *Theorem 11 does not provide an explicit bound on the difference between* $\mathbb{E}\left[D_N^{true}\right]$ *and* D_{appr}^{true}*, defined via* (122), (128)*, respectively. This deficiency will be partially remedied in Section 5.1, via numerical experiments.*

4.5.2 Distances to points in a given quadrant

In this section, we study the distances from a fixed vector $\boldsymbol{a} \in \mathbb{R}^d$ to standard normal random vectors, conditioned on $Q_{\boldsymbol{\sigma}}^d$ (see (6) in Section 2.1 and Definition 4 in Section 2.3). The results of this section will be used to analyze the quantities D_i^{susp} and prop_i, defined by (119), (120), respectively.

Throughout this section, we use the following definition.

Definition 9. *Suppose that* $d, N > 0$ *are positive integers, and that* $\boldsymbol{a}, \boldsymbol{x}_1, \dots, \boldsymbol{x}_N$ *are vectors in* $\mathbb{R}^d$*. For* $j = 1, \dots, N$*, we define* $\boldsymbol{x}_{t(\boldsymbol{a},j)}$ *to be the* j*th nearest neighbor of* $\boldsymbol{a}$ *among* $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$*. In other words,*

$$\|\boldsymbol{x}_{t(\boldsymbol{a},1)} - \boldsymbol{a}\|^2 \leq \dots \leq \|\boldsymbol{x}_{t(\boldsymbol{a},N)} - \boldsymbol{a}\|^2. \tag{137}$$

Next, we prove a number of technical lemmas.

Lemma 1. *Suppose that* $N > 0$ *and* $d \geq L > 0$ *are positive integers, and that* $\boldsymbol{a} \in \mathbb{R}^d$ *is a vector, all of whose coordinates are positive. Suppose further that* $\boldsymbol{\sigma}$ *is a word of symbols* $+, -$ *of length* L*, and that* $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$ *are i.i.d. random vectors in* $\mathbb{R}^d$*, such that for all* $i = 1, \dots, N$*,*

$$\boldsymbol{x}_i \sim N\left(0_d, I_d\right) \mid Q_{\boldsymbol{\sigma}}^d \tag{138}$$

(see (6) *in Section 2.1 and Definition 4 in Section 2.3). Then, for all* $j = 1, \dots, N$*, the expectation of the square of the distance from* $\boldsymbol{a}$ *to* $\boldsymbol{x}_{t(\boldsymbol{a},j)}$ *(see Definition 9 above) is given by the formula*

$$\mathbb{E}\left[\|\boldsymbol{x}_{t(\boldsymbol{a},j)} - \boldsymbol{a}\|^2\right] = \int_0^1 (F_{\boldsymbol{a}}^{\boldsymbol{\sigma}})^{-1}(t) \cdot f_{B(j,N+1-j)}(t)\, dt, \tag{139}$$

where $(F_{\boldsymbol{a}}^{\boldsymbol{\sigma}})^{-1}$ *is the inverse of the function* $F_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$ *defined via* (94)*, and* $f_{B(j,N+1-j)}$ *is defined via* (28)*.*

Proof. For all $i = 1, \dots, N$, we define the random variable D_i by the formula

$$D_i = \|\boldsymbol{x}_i - \boldsymbol{a}\|^2. \tag{140}$$

In other words, D_i is the square of the distance from $\boldsymbol{x}_i$ to $\boldsymbol{a}$. Then, $D_1, \dots, D_N$ are i.i.d., and for all $i = 1, \dots, N$

$$D_i \sim D_{\boldsymbol{a}}^{\boldsymbol{\sigma}}, \tag{141}$$

due to Definition 8 in Section 3. Next, due to Theorem 9 in Section 3, the cdf of any D_i is given by $F_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$, defined via (94). Thus the identity (139) follows from Theorem 5 in Section 3. ■

Corollary 2. *Under the hypothesis of Lemma 1, for $j = 1, \ldots, N/2$,*

$$\mathbb{E}\left[\|\boldsymbol{x}_{t(\boldsymbol{a},j)} - \boldsymbol{a}\|^2\right] = (F_{\boldsymbol{a}}^{\boldsymbol{\sigma}})^{-1}\left(\frac{j}{N+1}\right) + O\left(d + \|\boldsymbol{a}\|^2\right), \tag{142}$$

and for $j = N/2, \ldots, N$,

$$\mathbb{E}\left[\|\boldsymbol{x}_{t(\boldsymbol{a},j)} - \boldsymbol{a}\|^2\right] = (F_{\boldsymbol{a}}^{\boldsymbol{\sigma}})^{-1}\left(\frac{j}{N+1}\right) + O\left(j \cdot \left(d + \|\boldsymbol{a}\|^2\right)\right). \tag{143}$$

In other words, both formulae (142) *and* (143) *hold, if we fix k, L and let $d \to \infty$.*

Proof. Obviously, for all $i = 1, \ldots, N$

$$\mathbb{E}\left[\|\boldsymbol{x}_i - \boldsymbol{a}\|^2\right] = O\left(\mathbb{E}\left[\chi^2\left(d, \|\boldsymbol{a}\|^2\right)\right]\right) = O\left(d + \|\boldsymbol{a}\|^2\right). \tag{144}$$

Therefore, (142), (143) follow from the combination of Theorem 1 in Section 2.3 and (139) in essentially the same way, as (134) was derived from (127) in the proof of Theorem 11 in Section 4.5.1. ■

The remainder of this section is devoted to generalizing Lemma 1 and Corollary 2 to standard normal random vectors, conditioned on a union of several quadrants. First, we introduce the following definition.

Definition 10. *Suppose that $d \geq L > 0$ are positive integers, and that $\boldsymbol{\sigma}$ is a word of symbols $+, -$ of length L. Suppose further, that $\boldsymbol{a} \in \mathbb{R}^d$ is a vector, all of whose coordinates are positive. We define the function $G_{\boldsymbol{a}}^{\boldsymbol{\sigma}} : \mathbb{R} \to (0,1)$ by the formula*

$$G_{\boldsymbol{a}}^{\boldsymbol{\sigma}}(x) = \frac{1}{L+1}\sum_{j=0}^{L} F_{\boldsymbol{a}}^{\boldsymbol{\sigma}^j}(x), \tag{145}$$

where $\boldsymbol{\sigma}^0, \ldots, \boldsymbol{\sigma}^L$ are as in Definition 1 in Section 2.1, and $F_{\boldsymbol{a}}^{\boldsymbol{\sigma}^j}$ is defined via (94) *in Section 3. In other words, $G_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$ is the average of $L+1$ functions $F_{\boldsymbol{a}}^{\boldsymbol{\mu}}$, taken over those words $\boldsymbol{\mu}$, whose degree of contact with $\boldsymbol{\sigma}$ is either zero or one. Obviously, $G_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$ retains all the properties of the cdf of a continuous distribution.*

The following lemma generalizes Lemma 1.

Lemma 2. *Suppose that $N > 0$ and $d \geq L > 0$ are positive integers, and that $\boldsymbol{a} \in \mathbb{R}^d$ is a vector, all of whose coordinates are positive. Suppose further that $\boldsymbol{\sigma}$ is a word of symbols $+, -$ of length L, and that $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ are i.i.d. random vectors in $\mathbb{R}^d$, and for all $i = 1, \ldots, N$,*

$$\boldsymbol{x}_i \sim N\left(0_d, I_d\right) \mid Q_{\boldsymbol{\sigma}^0}^d \cup \cdots \cup Q_{\boldsymbol{\sigma}^L}^d \tag{146}$$

(see (5), (6) *in Section 2.1 and Definition 4 in Section 2.3). Then, for all $j = 1, \ldots, N$, the expectation of the square of the distance from $\boldsymbol{a}$ to $\boldsymbol{x}_{t(\boldsymbol{a},j)}$ (see Definition 9 in Section 4.5.2) is given by the formula*

$$\mathbb{E}\left[\|\boldsymbol{x}_{t(\boldsymbol{a},j)} - \boldsymbol{a}\|^2\right] = \int_0^1 (G_{\boldsymbol{a}}^{\boldsymbol{\sigma}})^{-1}(t) \cdot f_{B(j,N+1-j)}(t)\, dt, \tag{147}$$

where $(G_{\boldsymbol{a}}^{\boldsymbol{\sigma}})^{-1}$ is the inverse of the function $G_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$ defined via (145)*, and $f_{B(j,N+1-j)}$ is defined via* (28)*.*

Proof. Since the standard normal distribution is radially symmetric, the probability of $\boldsymbol{x}_i$ being in $Q^d_{\boldsymbol{\sigma}^j}$ for any $i = 1, \ldots, N$ and $j = 0, \ldots, L$ are identical, i.e.

$$\mathbb{P}\left\{\boldsymbol{x}_i \in Q^d_{\boldsymbol{\sigma}^j}\right\} = \frac{1}{L+1}. \tag{148}$$

We combine (94) with (148) to conclude that the cdf of $\|\boldsymbol{x}_i - \boldsymbol{a}\|^2$ is given by $G^{\boldsymbol{\sigma}}_{\boldsymbol{a}}$, defined via (145). Thus the identity (147) follows from Theorem 5 in Section 3. ■

Corollary 3. *Under the hypothesis of Lemma 2, for* $j = 1, \ldots, N/2$*,*

$$\mathbb{E}\left[\|\boldsymbol{x}_{t(\boldsymbol{a},j)} - \boldsymbol{a}\|^2\right] = (G^{\boldsymbol{\sigma}}_{\boldsymbol{a}})^{-1}\left(\frac{j}{N+1}\right) + O\left(d + \|\boldsymbol{a}\|^2\right), \tag{149}$$

and for $j = N/2, \ldots, N$*,*

$$\mathbb{E}\left[\|\boldsymbol{x}_{t(\boldsymbol{a},j)} - \boldsymbol{a}\|^2\right] = (G^{\boldsymbol{\sigma}}_{\boldsymbol{a}})^{-1}\left(\frac{j}{N+1}\right) + O\left(j \cdot \left(d + \|\boldsymbol{a}\|^2\right)\right). \tag{150}$$

In other words, both formulae (149) *and* (150) *hold, if we fix* k, L *and let* $d \to \infty$*.*

Proof. The proof is essentially identical to the proof of Corollary 2. ■

4.5.3 Average distance to suspects

In this section, we study the expectation of D^{susp}_i, defined via (119). Obviously, it does not depend on i. Therefore, it suffices to compute $\mathbb{E}\,[D^{susp}_i]$ for $i = N$ only.

Lemma 3. *Suppose that* $k > 0$ *and* $d \geq L > 0$ *are positive integers, and that* N *is defined via* (121)*. Suppose further that* $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N-1}$ *are i.i.d. standard normal random vectors in* $\mathbb{R}^d$*, and that* $\boldsymbol{\sigma} = + \cdots +$ *is a word of length* L*. We define the integer random variable* J *to be the number of* $\boldsymbol{x}_i$*'s that belong to one of the quadrants* $Q^d_{\boldsymbol{\sigma}^0}, \ldots, Q^d_{\boldsymbol{\sigma}^L}$*, defined via* (5)*,* (6) *in Section 2.1. In other words,*

$$J = \left|\left\{\boldsymbol{x}_i \ : \ \boldsymbol{x}_i \in Q^d_{\boldsymbol{\sigma}^0} \cup \cdots \cup Q^d_{\boldsymbol{\sigma}^L}, \ \ i = 1, \ldots, N-1\right\}\right|. \tag{151}$$

Then, J *has binomial distribution* $Bin(N-1, p)$*, with real parameter* $0 < p < 1$*, defined by the formula*

$$p = \frac{L+1}{2^L}. \tag{152}$$

The binomial distribution is defined via (19) *in Section 2.3.*

Proof. Due to radial symmetry of standard normal distribution, the probability of any $\boldsymbol{x}_i$ being in $Q^d_{\boldsymbol{\mu}}$ does not depend on $\boldsymbol{\mu}$. Therefore, the probability of any $\boldsymbol{x}_i$ being in one of the quadrants $Q^d_{\boldsymbol{\sigma}^0}, \ldots, Q^d_{\boldsymbol{\sigma}^L}$ is given by p, defined via (152). Thus the random variable J, defined via (151), has distribution $Bin(N-1, p)$. ■

Corollary 4. *Under the assumptions of Lemma 3, the expectation and variance of J are given by the formulae*

$$\mathbb{E}\left[J\right] = k \cdot (L+1) \cdot \left(1 - \frac{1}{k \cdot 2^L}\right) = k \cdot (L+1) + O\left(2^{-L}\right) \tag{153}$$

and

$$\mathrm{Var}\left[J\right] = \mathbb{E}\left[J\right] \cdot \left(1 - \frac{L+1}{2^L}\right) = k \cdot (L+1) + O\left(L^2 \cdot 2^{-L}\right). \tag{154}$$

Proof. The identities (153), (154) follow from the combination of (20), (121) and Lemma 3. ■

Corollary 5. *Under the assumptions of Lemma 3, the probability of J being less that k is exponentially small in L and k. More precisely,*

$$\mathbb{P}\ \{J < k\} = O\left(\exp\left[-\frac{kL}{2}\right]\right). \tag{155}$$

Proof. Due to the central limit theorem and (153), (154), the probability of $J < k$ is given by the formula

$$\begin{aligned}\mathbb{P}\ \{J < k\} &\approx \mathbb{P}\left\{\frac{J - kL}{\sqrt{kL}} < -\sqrt{kL}\right\} \approx \Phi(-\sqrt{kL}) \\ &= O\left(\mathrm{erfc}\left(\sqrt{\frac{kL}{2}}\right)\right) = O\left(\exp\left[-\frac{kL}{2}\right]\right),\end{aligned} \tag{156}$$

where the functions erfc, Φ are given by (13), (22), respectively. ■

Theorem 12. *Suppose that $k > 0$ and $d \geq L > 0$ are positive integers, N is defined via* (121)*, and $\boldsymbol{a} \in \mathbb{R}^d$ is a vector, all of whose coordinates are positive. Suppose further that $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ are i.i.d. standard normal random vectors in $\mathbb{R}^d$, and D_N^{susp} is defined via* (119)*. Then, the conditional expectation of D_N^{susp} given that $\boldsymbol{x}_N = \boldsymbol{a}$ is provided by the formula*

$$\begin{aligned}&\mathbb{E}\left[D_N^{susp} \mid \boldsymbol{x}_N = \boldsymbol{a}\right] = \\ &\frac{1}{k}\sum_{j=k}^{N-1} \mathbb{P}\ \{J = j\} \cdot \sum_{i=1}^{k} \int_0^1 \left(G_{\boldsymbol{a}}^{\boldsymbol{\sigma}}\right)^{-1}(t) \cdot f_{B(i,j+1-i)}(t)\, dt,\end{aligned} \tag{157}$$

where J is defined via (151)*, $\boldsymbol{\sigma} = + \cdots +$ is a word of length L, $\left(G_{\boldsymbol{a}}^{\boldsymbol{\sigma}}\right)^{-1}$ is the inverse of the function $G_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$ defined via* (145)*, and $f_{B(i,j+1-i)}$ is the pdf of the beta distribution, defined via* (28)*.*

Proof. Suppose that $\boldsymbol{x}_N = \boldsymbol{a}$. In particular, $\boldsymbol{x}_N \in B_{\boldsymbol{\sigma}}$ for $\boldsymbol{\sigma} = +, \ldots, +$, as defined by (100). Then, the k suspects $\boldsymbol{x}_{s(N,1)}, \ldots, \boldsymbol{x}_{s(N,k)}$ of $\boldsymbol{x}_N$ are chosen among those points $\boldsymbol{x}_i$ with $i = 1, \ldots, N-1$, that belong to the set $V_{\boldsymbol{\sigma}}$, defined via (107). We denote the indices of these points by $c(1,J), \cdots, c(J,J)$. In other words,

$$V_{\boldsymbol{\sigma}} = \left\{\boldsymbol{x}_{c(1,J)}, \ldots, \boldsymbol{x}_{c(J,J)}\right\}. \tag{158}$$

The distribution of the number of points J in $V_{\boldsymbol{\sigma}}$ is described in Lemma 3 in Section 4.5.3.

Now, suppose that $J = j$ for some $j = k, \dots, N-1$. In this case, the vectors $\boldsymbol{x}_{c(1,j)}, \dots, \boldsymbol{x}_{c(j,j)}$ are i.i.d., and, for $i = 1, \dots, j$,

$$\boldsymbol{x}_{c(i,j)} \sim N(0_d, I_d) \mid Q^d_{\boldsymbol{\sigma}^0} \cup \cdots \cup Q^d_{\boldsymbol{\sigma}^L}, \tag{159}$$

as in (146). We combine Lemma 2, Lemma 3 and (159) to conclude that for $j = k, \dots, N-1$,

$$\mathbb{E}\left[D_N^{susp} \mid \boldsymbol{x}_N = \boldsymbol{a},\ J = j\right] = \frac{1}{k} \sum_{i=1}^{k} \int_0^1 (G^{\boldsymbol{\sigma}}_{\boldsymbol{a}})^{-1}(t) \cdot f_{B(i,j+1-i)}(t)\ dt. \tag{160}$$

If $j = 1, \dots, k-1$, we say that $D_N^{susp} = 0$. Due to Corollary 5, the probability of the event $\{J < k\}$ is exponentially small, hence the effects of this convention are negligible. Obviously,

$$\mathbb{E}\left[D_N^{susp} \mid \boldsymbol{x}_N = \boldsymbol{a}\right] = \sum_{j=k}^{N-1} \mathbb{P}\ \{J = j\} \cdot \mathbb{E}\left[D_N^{susp} \mid \boldsymbol{x}_N = \boldsymbol{a},\ J = j\right]. \tag{161}$$

Thus the identity (157) follows from the combination of Lemma 3, (160) and (161). ■

The following theorem provides an approximation to the right-hand side of (157).

Theorem 13. *Suppose that $k > 0$ and $d \geq L > 0$ are positive integers, and N is defined via* (121)*. Suppose further that $\boldsymbol{\sigma} = + \cdots +$ is a word of length L. We define the function $D^{susp} : (0, \infty)^d \to \mathbb{R}$ by the formula*

$$D^{susp}(\boldsymbol{a}) = \frac{1}{k} \sum_{i=1}^{k} \left(G^{\boldsymbol{\sigma}}_{\boldsymbol{a}}\right)^{-1} \left(\frac{i}{k \cdot (L+1) + 1}\right), \tag{162}$$

where $\left(G^{\boldsymbol{\sigma}}_{\boldsymbol{a}}\right)^{-1}$ is the inverse of the function $G^{\boldsymbol{\sigma}}_{\boldsymbol{a}}$, defined via (145)*. Then, for all vectors with positive coordinates $\boldsymbol{a} \in (0, \infty)^d$,*

$$\mathbb{E}\left[D_N^{susp} \mid \boldsymbol{x}_N = \boldsymbol{a}\right] = D^{susp}(\boldsymbol{a}) + O\left(d + \|\boldsymbol{a}\|^2\right), \quad d \to \infty. \tag{163}$$

where $\mathbb{E}\left[D_N^{susp} \mid \boldsymbol{x}_N = \boldsymbol{a}\right]$ is the same as in (157) *of Theorem 12. Also,*

$$\mathbb{E}\left[D_N^{susp} \mid \boldsymbol{x}_N = \boldsymbol{a}\right] = D^{susp}(\boldsymbol{a}) + O\left(\frac{1}{\sqrt{k}}\right), \quad k \to \infty. \tag{164}$$

In other words, the identity (163) *holds, if we keep k, L fixed and let $d \to \infty$, and the identity* (164) *holds, if we keep d, L fixed and let $k \to \infty$.*

Proof. We combine Lemma 2, Corollary 2 in Section 4.5.2 and Theorem 12 to conclude that

$$\mathbb{E}\left[D_N^{susp} \mid \boldsymbol{x}_N = \boldsymbol{a}\right] = \\ \frac{1}{k} \cdot \sum_{j=k}^{N-1} \mathbb{P}\ \{J = j\} \cdot \sum_{i=1}^{k} (G^{\boldsymbol{\sigma}}_{\boldsymbol{a}})^{-1} \left(\frac{i}{j+1}\right) + O\left(d + \|\boldsymbol{a}\|^2\right). \tag{165}$$

By carrying out manipulations along the lines of the proofs of Theorem 11 in Section 4.5.1 and Corollaries 2, 3 in Section 4.5.2, we easily see that

$$\mathbb{E}\left[D_N^{susp} \mid \boldsymbol{x}_N = \boldsymbol{a}\right] = O\left(d + \|\boldsymbol{a}\|^2\right). \tag{166}$$

Thus (163) follows from the combination of Corollary 4, Corollary 5, (165) and (166). The proof of (164) using Theorem 6 in Section 3 is analogous to the proof of (130) in Theorem 11 in Section 4.5.1. ■

The following theorem provides an approximation to the expectation $\mathbb{E}\left[D_N^{susp}\right]$, defined via (119). The error of this approximation will be verified via numerical experiments.

Theorem 14. *Suppose that $k > 0$ and $d \geq L > 0$ are positive integer, and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ are i.i.d. standard normal random vectors in $\mathbb{R}^d$. We define the real number D_{appr}^{susp} by the formula*

$$D_{appr}^{susp} = \int_{\lambda=0}^{\infty} \operatorname*{Avg}_{S_d^+(\sqrt{\lambda})} \left(D^{susp}(\boldsymbol{a})\right) \cdot f_{\chi_d^2}(\lambda)\, d\lambda, \tag{167}$$

where the function $f_{\chi_d^2}$ is defined via (23), the set $S_d^+(\sqrt{\lambda})$ is defined via (11), the function $D^{susp} : (0,\infty)^d \to \mathbb{R}$ is defined via (162), and the average of D^{susp} over $S_d^+(\sqrt{\lambda})$ is taken in the sense of Definition 2 in Section 2.2 with respect to the $(d-1)$-dimensional area integral. Then,

$$\mathbb{E}\left[D_N^{susp}\right] = D_{appr}^{susp} + O(d), \quad d \to \infty. \tag{168}$$

where the real random variable D_N^{susp} is defined via (119). Also,

$$\mathbb{E}\left[D_N^{susp}\right] = D_{appr}^{susp} + O\left(\frac{1}{\sqrt{k}}\right), \quad k \to \infty. \tag{169}$$

In other words, (168) holds, if we fix k, L and let $d \to \infty$, and (169) holds, if we fix d, L and let $k \to \infty$.

Proof. We define $\boldsymbol{\sigma} = +, \ldots, +$ to be the word of length L and observe that

$$Q_{\boldsymbol{\sigma}}^d = (0,\infty)^d, \tag{170}$$

due to (6). Obviously, due to radial symmetry of the standard normal distribution,

$$\mathbb{E}\left[D_N^{susp}\right] = \int_{Q_{\boldsymbol{\sigma}}^d} \mathbb{E}\left[D_N^{susp} \mid \boldsymbol{x}_N = \boldsymbol{a}\right] \cdot \mathbb{P}\ \{\boldsymbol{x}_N = \boldsymbol{a}\}. \tag{171}$$

We substitute (163) into (171) to obtain the formula

$$\mathbb{E}\left[D_N^{susp}\right] = \int_{Q_{\boldsymbol{\sigma}}^d} \left(D^{susp}(\boldsymbol{a}) + O\left(d + \|\boldsymbol{a}\|^2\right)\right) \cdot \mathbb{P}\ \{\boldsymbol{x}_N = \boldsymbol{a}\}. \tag{172}$$

We combine (23) and (24) to conclude that

$$\int_{Q^d_{\boldsymbol{\sigma}}} O\left(d + \|\boldsymbol{a}\|^2\right) \cdot \ \mathbb{P}\ \{\boldsymbol{x}_N = \boldsymbol{a}\} = O(d). \tag{173}$$

Next, we combine (18), (10) and (11) in Section 2.2 to compute

$$\begin{aligned}
&\int_{Q^d_{\boldsymbol{\sigma}}} D^{susp}(\boldsymbol{a})\ \mathbb{P}\ \{\boldsymbol{x}_N = \boldsymbol{a}\} = \\
&\int_{Q^d_{\boldsymbol{\sigma}}} \left(\frac{2}{\pi}\right)^{d/2} e^{-\|\boldsymbol{a}\|^2/2} \cdot D^{susp}(\boldsymbol{a})\ d\boldsymbol{a} = \\
&\left(\frac{2}{\pi}\right)^{d/2} \int_{r=0}^{\infty} e^{-r^2/2} \int_{S^+(r)} D^{susp}(\boldsymbol{a})\ d\Omega\ dr = \\
&\left(\frac{2}{\pi}\right)^{d/2} \int_{\lambda=0}^{\infty} \frac{d\lambda}{2\sqrt{\lambda}} e^{-\lambda/2} \cdot \mathrm{Area}(S^+_d(\sqrt{\lambda})) \cdot \underset{S^+_d(\sqrt{\lambda})}{\mathrm{Avg}} \left(D^{susp}(\boldsymbol{a})\right) = \\
&\int_{\lambda=0}^{\infty} \frac{\lambda^{d/2-1} e^{-\lambda/2}}{2^{d/2}\Gamma(d/2)} \cdot \underset{S^+_d(\sqrt{\lambda})}{\mathrm{Avg}} \left(D^{susp}(\boldsymbol{a})\right)\ d\lambda.
\end{aligned} \tag{174}$$

Finally, to prove the identity (168), we combine (23), (172), (173) and (174). The identity (169) readily follows from the combination of (164) and (174). ■

Remark 4. *Clearly, one can substitute* (157) *into* (171) *and carry out manipulations along the lines of* (174)*, to obtain an exact formula for* $\mathbb{E}\left[D^{susp}_N\right]$*. Theorem 14 does not provide an explicit bound on the difference between* $\mathbb{E}\left[D^{susp}_N\right]$ *and* D^{susp}_{appr}*, defined via* (167)*. This deficiency will be partially remedied in Section 5.1, via numerical experiments.*

4.5.4 Proportion of suspects among true nearest neighbors

In this section, we study the expectation of prop_i, defined via (120). Obviously, it does not depend on i. Therefore, it suffices to compute $\mathbb{E}\left[\mathrm{prop}_i\right]$ for $i = N$ only.

First, we need to prove a number of technical lemmas.

Lemma 4. *Suppose that* $L > 0$ *is a positive integer, and that the function* $F : \mathbb{R} \to [0, 1]$ *is the cdf of a positive real continuous random variables. For all integer* $k > 0$*, we define* N *via* (121)*, and also define the random variable* $\hat{F}_N(t)$ *for all real* t *via* (50) *in Section 2.3. Then,*

$$\mathbb{E}\left[\hat{F}_N\left(F^{-1}(2^{-L})\right)\right] = 2^{-L} + O\left(\frac{1}{\sqrt{k}}\right). \tag{175}$$

In other words, if $X_1, \ldots, X_N$ *are i.i.d. random variables with cdf* F*, then the expected proportion of* X_i*'s below* $F^{-1}(2^{-L})$ *is* k/N *up to an error of order* $O(k^{-1/2})$*. Put differently, the smallest* k *values* $X_{(1)}, \ldots, X_{(k)}$ *are expected to lie in the interval*

$$I = \left(0, F^{-1}(2^{-L}) + O\left(\frac{1}{\sqrt{k}}\right)\right). \tag{176}$$

Proof. The identity (175) follows from the combination of Theorems 3, 4 in Section 2.3. ■

Lemma 5. *Suppose that $L > 0$ is a positive integer, and that $\alpha > 0$ is a positive real number. Suppose further, that the function $G : \mathbb{R} \to [0,1]$ is the cdf of a positive real random variable. For all positive integers $k > 0$, we define positive integer $n_1 = n_1(k)$ by the formula*

$$n_1 = k \cdot (L+1) + \lfloor \alpha \cdot \sqrt{k \cdot L} \rfloor. \tag{177}$$

Also, we define the random variable $\hat{G}_{n_1}(t)$ for all real t via (50) in Section 2.3. Then,

$$\mathbb{E}\left[\hat{G}_{n_1}(t)\right] = G(t) + O\left(\frac{1}{\sqrt{k \cdot L}}\right). \tag{178}$$

Proof. The identity (178) follows from the combination of Theorems 3, 4 in Section 2.3. ■

Theorem 15. *Suppose that $k > 0$ and $d \geq L > 0$ are positive integers, and that N is defined via (121). Suppose further that $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ are i.i.d. standard normal random vectors in $\mathbb{R}^d$, and $\boldsymbol{\sigma} = + \cdots +$ is a word of length L. We define the function $P : (0,\infty)^d \to [0,1]$ by the formula*

$$P(\boldsymbol{a}) = (L+1) \cdot G_{\boldsymbol{a}}^{\boldsymbol{\sigma}}\left(F^{-1}_{\chi^2(d,\|\boldsymbol{a}\|^2)}(2^{-L})\right), \tag{179}$$

where the function $F^{-1}_{\chi^2(d,\|\boldsymbol{a}\|^2)}$ is the inverse of the cdf of $\chi^2(d, \|\boldsymbol{a}\|^2)$, defined via (25) in Section 2.3, and the function $G_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$ is defined via (145). Also, we define the random variable prop_N *via (120). Then, the conditional expectation of* prop_N *given that $\boldsymbol{x}_N = \boldsymbol{a}$ is provided by the formula*

$$\mathbb{E}\left[\text{prop}_N \mid \boldsymbol{x}_N = \boldsymbol{a}\right] = P(\boldsymbol{a}) + O\left(\sqrt{\frac{L}{k}}\right), \tag{180}$$

for any vector $\boldsymbol{a} \in \mathbb{R}^d$, all of whose coordinates are positive.

Proof. Suppose that $\boldsymbol{x}_N = \boldsymbol{a}$. In particular, $\boldsymbol{x}_N \in B_{\boldsymbol{\sigma}}$ for $\boldsymbol{\sigma} = +, \ldots, +$, as defined by (100). Then, the k suspects $\boldsymbol{x}_{s(N,1)}, \ldots, \boldsymbol{x}_{s(N,k)}$ of $\boldsymbol{x}_N$ are chosen among those points $\boldsymbol{x}_i$ with $i = 1, \ldots, N-1$, that belong to the set $V_{\boldsymbol{\sigma}}$, defined via (107). As in Lemma 3 in Section 4.5.3, we define the random variable J to be the number of points in $V_{\boldsymbol{\sigma}}$. Due to Corollary 4 in Section 4.5.3,

$$\frac{J}{k} = (L+1) + O_p\left(\sqrt{\frac{L}{k}}\right), \quad k \to \infty, \tag{181}$$

in the sense of Definition 6 in Section 2.3. In other words, J/k converges in probability to $(L+1)$, as $k \to \infty$, and the error is of order $O(k^{-1/2})$. Next, for any real positive number $r > 0$, we define $\gamma(r)$ by the formula

$$\gamma(r) = \frac{1}{J} \cdot \left|\left\{\boldsymbol{x} \in V_{\boldsymbol{\sigma}} \ : \ \|\boldsymbol{x} - \boldsymbol{a}\|^2 < r\right\}\right|. \tag{182}$$

In other words, $0 \le \gamma(r) \le 1$ is the proportion of those points $\boldsymbol{x} \in V_{\boldsymbol{\sigma}}$, whose distance to $\boldsymbol{a}$ is at most $\sqrt{r}$.

We combine (181), (182), Definition 10 in Section 4.5.2, Lemma 3, Corollary 4 in Section 4.5.3 and Lemma 5 to conclude that

$$\mathbb{E}\left[\frac{J}{k} \cdot \gamma(r) \mid \boldsymbol{x}_N = \boldsymbol{a}\right] = (L+1) \cdot G_{\boldsymbol{a}}^{\boldsymbol{\sigma}}(r) + O\left(\sqrt{\frac{L}{k}}\right). \tag{183}$$

Thus (180) follows from the combination of (183) and Lemma 4. ■

We conclude the section with the theorem, that provides an approximation to the expectation of prop_N, defined via (120).

Theorem 16. *Suppose that $k > 0$ and $d \ge L > 0$ are positive integer, and $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$ are i.i.d. standard normal random vectors in $\mathbb{R}^d$. We define the real number P_{appr} by the formula*

$$P_{appr} = \int_{\lambda=0}^{\infty} \underset{S_d^+(\sqrt{\lambda})}{\text{Avg}} \left(P(\boldsymbol{a})\right) \cdot f_{\chi_d^2}(\lambda)\, d\lambda, \tag{184}$$

where the function $f_{\chi_d^2}$ is defined via (23), the set $S_d^+(\sqrt{\lambda})$ is defined via (11), the function $P : (0,\infty)^d \to \mathbb{R}$ is defined via (179), and the average of P over $S_d^+(\sqrt{\lambda})$ is taken in the sense of Definition 2 with respect to the $(d-1)$-dimensional area integral. Then,

$$\mathbb{E}\left[\text{prop}_N\right] = P_{appr} + O\left(\sqrt{\frac{L}{k}}\right), \tag{185}$$

where the real random variable prop_N *is defined via* (120)*. In other words,* (185) *holds, if we fix d, L and let $k \to \infty$.*

Proof. The identity (185) follows from the combination of Theorem 15 and manipulations along the lines of the proof of Theorem 14. ■

Remark 5. *Theorem 16 does not provide an explicit estimate on the accuracy of* (185)*. This deficiency will be partially remedied in Section 5.1, via numerical experiments.*

5 Numerical Results

This section has two principal purposes. First, we demonstrate the performance of the algorithm described in Sections 4.2, 4.3. Second, we numerically evaluate the formulae developed in Section 4.5 and compare the results to the output of the algorithm.

Both algorithm and the computations have been implemented in FORTRAN (Lahey 95 Linux version). MATLAB and Mathematica have been used for some auxiliary tasks (e.g. graphics, symbolic manipulations etc.). The numerical experiments have been carried out on a modern laptop computer, with DualCore CPU 2.53 GHz and 2.9GB RAM.

5.1 Numerical illustration of the analysis

In this section, we illustrate the analysis of Section 4.5 via several numerical examples.

5.1.1 Experiment 1: distance to true nearest neighbors

In this experiment, we approximate $\mathbb{E}\left[D_N^{true}\right]$, defined via (118), by using three different schemes. The results of the computations are compared in Tables 2, 4. See also Figures 2, 3, 4(a).

In the computations, we first choose more or less arbitrarily the positive integer parameters d, k, L and set N via (121).

First way to approximate $\mathbb{E}\left[D_N^{true}\right]$: Monte Carlo. We generate N standard normal random vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$ in $\mathbb{R}^d$ and compute the quantity D_N^{true}, defined by (118). This procedure is repeated $M > 0$ times to yield a sequence of values $D_N^{true}(1), \dots, D_N^{true}(M)$, all computed via (118). For this sequence, we compute the sample mean

$$\mathbb{E}_{smpl}\left[D^{true}\right] = \frac{1}{M}\sum_{i=1}^{M} D_N^{true}(i) \tag{186}$$

and the sample variance

$$\mathrm{Var}_{smpl}\left[D^{true}\right] = \frac{1}{M}\sum_{i=1}^{M}\left(D_N^{true}(i) - \mathbb{E}_{smpl}\left[D^{true}\right]\right)^2. \tag{187}$$

Also, we define the statistical error by the formula

$$err_{stat} = \frac{2}{\mathbb{E}_{smpl}\left[D^{true}\right]} \cdot \sqrt{\frac{\mathrm{Var}_{smpl}\left[D^{true}\right]}{M}}. \tag{188}$$

This formula roughly gives the relative error of the estimation of $\mathbb{E}\left[D_N^{true}\right]$ (118) by the sample mean (186). The values (186), (188) appear in the last two columns of Tables 2, 4.

Second way to approximate $\mathbb{E}\left[D_N^{true}\right]$: Theorem 10. We evaluate the integral (122) numerically, by using the trapezoidal rule. The precision of this calculation is at least three decimal digits. The result appears in the third column of Tables 2, 4.

Third way to approximate $\mathbb{E}\left[D_N^{true}\right]$: Theorem 11. We evaluate the integral (128) numerically, by using the trapezoidal rule. The precision of this calculation is at least three decimal digits. The result appears in the fourth column of Tables 2, 4.

Structure of Tables 2, 4. The results of Experiment 1 are shown in Tables 2, 4 below. The first and second column contain the dimensionality d and the integer parameter L, respectively. The number of points N is set via (121), with $k = 30$. The third column contains the value $\mathbb{E}\left[D_N^{true}\right]$, computed via numerical evaluation of (122). The fourth column contains the value D_{appr}^{true}, computed via numerical evaluation of (128). The fifth column contains the relative error of approximating $\mathbb{E}\left[D_N^{true}\right]$ by D_{appr}^{true}. In other words,

$$err = \frac{\left|\mathbb{E}\left[D_N^{true}\right] - D_{appr}^{true}\right|}{\mathbb{E}\left[D_N^{true}\right]}. \tag{189}$$

The sixth column contains the sample mean $\mathbb{E}_{smpl}\left[D^{true}\right]$, evaluated via (186). The last column contains the relative statistical error, computed via (188).

Table 1 contains the choice of parameters k, L, N, d, M, corresponding to Table 2. For $L > 15$, the quantity (186) was not computed.

Table 3 contains the choice of parameters k, L, N, d, M, corresponding to Table 4. For $L > 15$, the quantity (186) was not computed.

k	d	L	N	M
30	$11, 20, 30, \ldots, 110$	10	30,720	10^5
30	$16, 20, 30, \ldots, 110$	15	983,040	10^4
30	$21, 25, 30, 40, \ldots, 110$	20	31,457,280	-
30	$26, 30, 40, \ldots, 110$	25	1,006,632,960	-

Table 1: *Parameters for Table 2 (see Section 5.1.1).*

5.1.2 Experiment 2: distance to suspects

In this experiment, we approximate $\mathbb{E}\left[D_N^{susp}\right]$, defined via (119), by using three different schemes. The results of the computations are compared in Tables 6, 8. See also Figures 2, 3, 4(a).

In the computations, we first choose more or less arbitrarily the positive integer parameters d, k, L and set N via (121).

First way to approximate $\mathbb{E}\left[D_N^{susp}\right]$: Monte Carlo. We choose the positive integer parameter $M > 0$ and perform the following operations:

1. Generate a random vector $\boldsymbol{a} \sim N(0_d, I_d) \mid (0,\infty)^d$. In other words, $\boldsymbol{a}$ is the standard normal vector in $\mathbb{R}^d$, conditioned on the set $(0,\infty)^d$ in the sense of Definition 4 in Section 2.3.

2. Generate a random variable $J \sim Binom\left(N-1, (L+1)\cdot 2^{-L}\right)$, as in Lemma 3 in Section 4.5.3.

3. Generate J i.i.d. random vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_J$, such that for all $i = 1, \ldots, J$,
$$\boldsymbol{x}_i \sim N(0_d, I_d) \mid Q^d_{\boldsymbol{\sigma}^0} \cup \cdots \cup Q^d_{\boldsymbol{\sigma}^L}, \tag{190}$$
where $\boldsymbol{\sigma} = + \cdots +$ is a word of length L, and the quadrants $Q^d_{\boldsymbol{\sigma}^j}$ with $j = 0, \ldots, L$ are defined by (5), (6) (see also Lemma 2 in Section 4.5.2).

4. Define the average square of the distance D_{avg} from $\boldsymbol{a}$ to its k nearest neighbors among $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ by the formula
$$D_{avg} = \frac{1}{k}\sum_{i=1}^{k} \|\boldsymbol{x}_{t(\boldsymbol{a},i)} - \boldsymbol{a}\|^2, \tag{191}$$
in agreement with Definition 9 in Section 4.5.2.

d	L	$\mathbb{E}\left[D_N^{true}\right]$	D_{appr}^{true}	err	$\mathbb{E}_{smpl}\left[D^{true}\right]$	err_{stat}
11	10	0.39062E+01	0.39411E+01	0.89284E-02	0.39080E+01	0.22695E-02
20	10	0.12435E+02	0.12519E+02	0.67221E-02	0.12437E+02	0.14446E-02
30	10	0.24141E+02	0.24275E+02	0.55572E-02	0.24159E+02	0.10980E-02
40	10	0.37135E+02	0.37316E+02	0.48885E-02	0.37163E+02	0.90592E-03
50	10	0.50957E+02	0.51183E+02	0.44471E-02	0.50996E+02	0.78771E-03
60	10	0.65368E+02	0.65638E+02	0.41298E-02	0.65463E+02	0.70710E-03
70	10	0.80227E+02	0.80539E+02	0.38886E-02	0.80340E+02	0.64615E-03
80	10	0.95440E+02	0.95792E+02	0.36976E-02	0.95551E+02	0.59362E-03
90	10	0.11094E+03	0.11133E+03	0.35417E-02	0.11105E+03	0.55385E-03
100	10	0.12668E+03	0.12711E+03	0.34115E-02	0.12683E+03	0.52273E-03
110	10	0.14263E+03	0.14310E+03	0.33008E-02	0.14279E+03	0.49427E-03
16	15	0.52416E+01	0.52801E+01	0.73572E-02	0.52455E+01	0.59071E-02
20	15	0.84454E+01	0.85007E+01	0.65421E-02	0.84811E+01	0.50512E-02
30	15	0.18138E+02	0.18235E+02	0.53219E-02	0.18119E+02	0.36405E-02
40	15	0.29399E+02	0.29535E+02	0.46406E-02	0.29450E+02	0.30694E-02
50	15	0.41678E+02	0.41853E+02	0.41999E-02	0.41703E+02	0.26266E-02
60	15	0.54688E+02	0.54901E+02	0.38879E-02	0.54785E+02	0.23223E-02
70	15	0.68254E+02	0.68503E+02	0.36533E-02	0.68416E+02	0.21472E-02
80	15	0.82261E+02	0.82546E+02	0.34694E-02	0.82342E+02	0.19335E-02
90	15	0.96629E+02	0.96950E+02	0.33204E-02	0.96904E+02	0.18334E-02
100	15	0.11129E+03	0.11165E+03	0.31969E-02	0.11138E+03	0.17133E-02
110	15	0.12622E+03	0.12661E+03	0.30923E-02	0.12646E+03	0.16136E-02
21	20	0.65680E+01	0.66092E+01	0.62731E-02		
25	20	0.96302E+01	0.96851E+01	0.57026E-02		
30	20	0.13944E+02	0.14016E+02	0.51775E-02		
40	20	0.23783E+02	0.23890E+02	0.44822E-02		
50	20	0.34789E+02	0.34929E+02	0.40390E-02		
60	20	0.46635E+02	0.46809E+02	0.37288E-02		
70	20	0.59125E+02	0.59332E+02	0.34977E-02		
80	20	0.72126E+02	0.72365E+02	0.33179E-02		
90	20	0.85546E+02	0.85818E+02	0.31731E-02		
100	20	0.99319E+02	0.99623E+02	0.30537E-02		
110	20	0.11339E+03	0.11372E+03	0.29531E-02		
26	25	0.78876E+01	0.79310E+01	0.55042E-02		
30	25	0.10858E+02	0.10913E+02	0.50860E-02		
40	25	0.19487E+02	0.19572E+02	0.43756E-02		
50	25	0.29394E+02	0.29510E+02	0.39276E-02		
60	25	0.40233E+02	0.40379E+02	0.36169E-02		
70	25	0.51788E+02	0.51963E+02	0.33873E-02		
80	25	0.63913E+02	0.64118E+02	0.32096E-02		
90	25	0.76508E+02	0.76742E+02	0.30674E-02		
100	25	0.89497E+02	0.89761E+02	0.29505E-02		
110	25	0.10282E+03	0.10311E+03	0.28525E-02		

Table 2: *Square of the distance to true nearest neighbors (see Section 5.1.1).*

k	d	L	N	M
30	$4, 5, \ldots, 15$	d	$30 \cdot 2^L$	10^5
30	$16, 17, 18, 19$	d	$30 \cdot 2^L$	-
30	$20, 25, \ldots, 100$	d	$30 \cdot 2^L$	-

Table 3: *Parameters for Table 4 (see Section 5.1.1).*

d	L	$\mathbb{E}\left[D_N^{true}\right]$	D_{appr}^{true}	err	$\mathbb{E}_{smpl}\left[D^{true}\right]$	err_{stat}
4	4	0.14533E+01	0.14668E+01	0.93041E-02	0.14497E+01	0.41584E-02
5	5	0.17470E+01	0.17653E+01	0.10475E-01	0.17477E+01	0.36375E-02
6	6	0.20354E+01	0.20571E+01	0.10666E-01	0.20425E+01	0.32999E-02
7	7	0.23201E+01	0.23444E+01	0.10467E-01	0.23192E+01	0.30233E-02
8	8	0.26017E+01	0.26280E+01	0.10117E-01	0.26008E+01	0.27950E-02
9	9	0.28810E+01	0.29090E+01	0.97180E-02	0.28805E+01	0.26246E-02
10	10	0.31582E+01	0.31876E+01	0.93144E-02	0.31570E+01	0.24719E-02
11	11	0.34338E+01	0.34644E+01	0.89257E-02	0.34370E+01	0.23502E-02
12	12	0.37079E+01	0.37396E+01	0.85599E-02	0.37079E+01	0.22385E-02
13	13	0.39808E+01	0.40135E+01	0.82191E-02	0.39818E+01	0.21449E-02
14	14	0.42527E+01	0.42863E+01	0.79034E-02	0.42553E+01	0.20704E-02
15	15	0.45235E+01	0.45580E+01	0.76113E-02	0.45300E+01	0.19793E-02
16	16	0.47936E+01	0.48288E+01	0.73413E-02		
17	17	0.50629E+01	0.50988E+01	0.70914E-02		
18	18	0.53316E+01	0.53681E+01	0.68599E-02		
19	19	0.55996E+01	0.56368E+01	0.66450E-02		
20	20	0.58671E+01	0.59049E+01	0.64452E-02		
25	25	0.71978E+01	0.72383E+01	0.56271E-02		
30	30	0.85201E+01	0.85629E+01	0.50296E-02		
35	35	0.98363E+01	0.98813E+01	0.45739E-02		
40	40	0.11148E+02	0.11195E+02	0.42161E-02		
45	45	0.12456E+02	0.12505E+02	0.39279E-02		
50	50	0.13761E+02	0.13812E+02	0.36910E-02		
55	55	0.15065E+02	0.15117E+02	0.34928E-02		
60	60	0.16366E+02	0.16420E+02	0.33247E-02		
65	65	0.17666E+02	0.17722E+02	0.31802E-02		
70	70	0.18964E+02	0.19022E+02	0.30548E-02		
75	75	0.20261E+02	0.20321E+02	0.29449E-02		
80	80	0.21557E+02	0.21619E+02	0.28478E-02		
85	85	0.22852E+02	0.22916E+02	0.27615E-02		
90	90	0.24147E+02	0.24212E+02	0.26841E-02		
95	95	0.25440E+02	0.25507E+02	0.26145E-02		
100	100	0.26733E+02	0.26802E+02	0.25513E-02		

Table 4: *Square of the distance to true nearest neighbors (see Section 5.1.1).*

We repeat this sequence of operations $M > 0$ times, to obtain M values $D_{avg}(1), \ldots, D_{avg}(M)$, all computed via (191). Next, we compute the sample mean

$$\mathbb{E}_{smpl}\left[D^{susp}\right] = \frac{1}{M} \sum_{i=1}^{M} D_{avg}(i) \tag{192}$$

and the sample variance

$$\mathrm{Var}_{smpl}\left[D^{susp}\right] = \frac{1}{M} \sum_{i=1}^{M} \left(D_{avg}(i) - \mathbb{E}_{smpl}\left[D^{susp}\right]\right)^2. \tag{193}$$

Also, we define the statistical error by the formula

$$err_{stat} = \frac{2}{\mathbb{E}_{smpl}\left[D^{susp}\right]} \cdot \sqrt{\frac{\mathrm{Var}_{smpl}\left[D^{susp}\right]}{M}}, \tag{194}$$

in complete analogy to (186), (187) and (188). The value (192) appears in the third column of Tables 6, 8.

Second way to approximate $\mathbb{E}\left[D_N^{susp}\right]$: RANN. We implement the Randomized Approximate Nearest Neighbor algorithm (see Sections 4.2, 4.3). Then, we choose the positive integer parameters $1 \le M_1 < N$ and $M_2 > 0$. Next, we generate N i.i.d. random vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, such that

$$\boldsymbol{x}_i \sim N(0_d, I_d), \tag{195}$$

for all $i = 1, \ldots, N$. For each $\boldsymbol{x}_i$, the algorithm finds its k suspects and, in particular, evaluates D_i^{susp} via (119). We define the average of D_i^{susp} over $i = 1, \ldots, M_1$ by the formula

$$D_{algo}^{susp} = \frac{1}{M_1} \sum_{i=1}^{M_1} D_i^{susp}. \tag{196}$$

Generation of the points and invocation of RANN are repeated M_2 times, to yield a sequence of average values $D_{algo}^{susp}(1), \ldots, D_{algo}^{susp}(M_2)$, all computed via (196). Next, we compute the sample mean

$$\mathbb{E}_{smpl}\left[D_{algo}\right] = \frac{1}{M_2} \sum_{i=1}^{M_2} D_{algo}^{susp}(i) \tag{197}$$

and the sample variance

$$\mathrm{Var}_{smpl}\left[D_{algo}\right] = \frac{1}{M_2} \sum_{i=1}^{M_2} \left(D_{algo}^{susp}(i) - \mathbb{E}_{smpl}\left[D_{algo}\right]\right)^2. \tag{198}$$

Also, we define the statistical error by the formula

$$err_{algo} = \frac{2}{\mathbb{E}_{smpl}\left[D_{algo}\right]} \cdot \sqrt{\frac{\mathrm{Var}_{smpl}\left[D_{algo}\right]}{M_2}}, \tag{199}$$

in complete analogy to (192), (193) and (194). The values (197), (199) appear in the fourth and fifth columns of Tables 6, 8, respectively.

Third way to approximate $\mathbb{E}\left[D_N^{susp}\right]$: Theorem 14. We choose the positive integer parameter $K > 0$ and evaluate numerically the quantity D_{appr}^{susp}, defined via (167). This evaluation consists of the following steps.

1. Choose equispaced discretization points $t_1, \dots, t_{50}$ of the interval
$$I = \left(10^{-3}, 1 - 10^{-3}\right). \tag{200}$$
2. Define $\lambda_1, \dots, \lambda_{50}$ by the formula
$$\lambda_i = F_{\chi_d^2}^{-1}(t_i), \tag{201}$$
for all $i = 1, \dots, 50$, where $F_{\chi_d^2}^{-1}$ is the inverse of the cdf of χ_d^2 distribution (see (23)).
3. For each $i = 1, \dots, 50$, generate K i.i.d. random vectors $\boldsymbol{a}_1^i, \dots, \boldsymbol{a}_K^i$, having uniform distribution on the set $S_d^+(\sqrt{\lambda_i})$, defined via (11).
4. For each $i = 1, \dots, 50$ and $j = 1, \dots, K$, evaluate $D^{susp}(\boldsymbol{a}_j^i)$, defined via (162). The evaluation is based on Theorem 9 in Section 3. More specifically, $G_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$ in (162) is computed via inverse Fourier transform of $h_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$ defined by (86), and $\left(G_{\boldsymbol{a}}^{\boldsymbol{\sigma}}\right)^{-1}$ is computed via a combination of bisection and Newton's method.
5. For each $i = 1, \dots, 50$, compute the empirical average
$$\mathrm{Avg}(\lambda_i) = \frac{1}{K} \sum_{j=1}^{K} D^{susp}(\boldsymbol{a}_j^i). \tag{202}$$
Obviously,
$$\underset{S_d^+(\sqrt{\lambda_i})}{\mathrm{Avg}} \left(D^{susp}(\boldsymbol{a})\right) = \mathrm{Avg}(\lambda_i) + O\left(\frac{1}{\sqrt{K}}\right). \tag{203}$$
6. Evaluate the integral (167) by using the trapezoidal rule with nodes (201) and function values (202), i.e. by the formula
$$D_{num}^{susp} = (t_2 - t_1) \cdot \left(\frac{1}{2}\left(\mathrm{Avg}(\lambda_1) + \mathrm{Avg}(\lambda_{50})\right) + \sum_{i=2}^{49} \mathrm{Avg}(\lambda_i)\right). \tag{204}$$

The value (204) appears in the sixth column of Tables 6, 8.

Structure of Tables 6, 8. The results of Experiment 2 are shown in Tables 6, 8 below. The first two columns contain the dimensionality d and the integer parameter L. The third column contains the sample mean $\mathbb{E}_{smpl}\left[D^{susp}\right]$, defined via (192). The fourth and fifth columns contain the sample mean $\mathbb{E}_{smpl}\left[D_{algo}\right]$ and its relative statistical error err_{algo}, defined via (197), (199), respectively. The sixth column contains D_{num}^{susp}, defined via

(204). The last column contains the relative error err_{num} of approximating $\mathbb{E}_{smpl}\left[D^{susp}\right]$ by D^{susp}_{num}, defined by the formula

$$err_{num} = \frac{|D^{susp}_{num} - \mathbb{E}_{smpl}\left[D^{susp}\right]|}{\mathbb{E}_{smpl}\left[D^{susp}\right]}. \tag{205}$$

Table 5 contains the choice of parameters $k, d, L, N, M, M_1, M_2, K$, corresponding to Table 6. Table 7 contains the choice of the same parameters, corresponding to Table 8. See also Figures 2, 3, 4(a).

k	d	L	N	M	M_1	M_2	K
30	$11, 20, 30, \ldots, 110$	10	30,720	10^5	100	1000	8000
30	$16, 20, 30, 40, 50$	15	983,040	10^5	100	100	8000
30	$60, 70, 80, 90, 100, 110$	15	983,040	10^5	-	-	8000
30	$21, 25, 30, 40, \ldots, 110$	20	31,457,280	-	-	-	8000
30	$26, 30, 40, \ldots, 110$	25	1,006,632,960	-	-	-	8000

Table 5: *Parameters for Table 6 (see Section 5.1.2).*

5.1.3 Experiment 3: proportion of suspects among true nearest neighbors

In this experiment, we approximate $\mathbb{E}\left[\text{prop}_N\right]$, defined via (120), by using two different schemes. The results of the computations are compared in Tables 10, 12. See also Figures 4(b), 5.

In the computations, we first choose more or less arbitrarily the positive integer parameters d, k, L and set N via (121).

First way to approximate $\mathbb{E}\left[\textbf{prop}_N\right]$: RANN. This computation is completely analogous to the second way to compute $\mathbb{E}\left[D^{susp}_N\right]$ in Experiment 2 (Section 5.1.2). More specifically, we implement the Randomized Approximate Nearest Neighbor algorithm (see Sections 4.2, 4.3). Then, we choose the positive integer parameters $1 \leq M_1 < N$ and $M_2 > 0$. Next, we generate N i.i.d. random vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, such that

$$\boldsymbol{x}_i \sim N(0_d, I_d), \tag{206}$$

for all $i = 1, \ldots, N$. For each $\boldsymbol{x}_i$, the algorithm finds its k suspects. For all $i = 1, \ldots, M_1$, we also find k true nearest neighbors of each $\boldsymbol{x}_i$ by direct scanning. Then, we compute prop_i via (120), for all $i = 1, \ldots, M_1$, and define the average of prop_{algo} by the formula

$$\text{prop}_{algo} = \frac{1}{M_1} \sum_{i=1}^{M_1} \text{prop}_i. \tag{207}$$

Generation of the points and invocation of RANN are repeated M_2 times, to yield a sequence of average values $\text{prop}_{algo}(1)$, ..., $\text{prop}_{algo}(M_2)$, all computed via (196). Next, we compute the sample mean

$$\mathbb{E}_{smpl}\left[\text{prop}_{algo}\right] = \frac{1}{M_2} \sum_{i=1}^{M_2} \text{prop}_{algo}(i) \tag{208}$$

d	L	$\mathbb{E}_{smpl}\left[D^{susp}\right]$	$\mathbb{E}_{smpl}\left[D_{algo}\right]$	err_{algo}	D^{susp}_{num}	err_{num}
11	10	0.48725E+01	0.48973E+01	0.22526E-02	0.48268E+01	0.93941E-02
20	10	0.16070E+02	0.16130E+02	0.13657E-02	0.15875E+02	0.12171E-01
30	10	0.30873E+02	0.30898E+02	0.10666E-02	0.30416E+02	0.14807E-01
40	10	0.46662E+02	0.46722E+02	0.94936E-03	0.45976E+02	0.14695E-01
50	10	0.63115E+02	0.63171E+02	0.79258E-03	0.62116E+02	0.15839E-01
60	10	0.79931E+02	0.79958E+02	0.70383E-03	0.78638E+02	0.16180E-01
70	10	0.97050E+02	0.97030E+02	0.65679E-03	0.95442E+02	0.16567E-01
80	10	0.11441E+03	0.11442E+03	0.58364E-03	0.11246E+03	0.17066E-01
90	10	0.13184E+03	0.13179E+03	0.60609E-03	0.12964E+03	0.16630E-01
100	10	0.14956E+03	0.14965E+03	0.57518E-03	0.14699E+03	0.17202E-01
110	10	0.16738E+03	0.16736E+03	0.54629E-03	0.16444E+03	0.17511E-01
16	15	0.70805E+01	0.71116E+01	0.56504E-02	0.69991E+01	0.11497E-01
20	15	0.11688E+02	0.11717E+02	0.43907E-02	0.11541E+02	0.12613E-01
30	15	0.25294E+02	0.25297E+02	0.39430E-02	0.24908E+02	0.15263E-01
40	15	0.40331E+02	0.40520E+02	0.28932E-02	0.39729E+02	0.14915E-01
50	15	0.56194E+02	0.56387E+02	0.27690E-02	0.55320E+02	0.15559E-01
60	15	0.72593E+02			0.71397E+02	0.16488E-01
70	15	0.89244E+02			0.87816E+02	0.15997E-01
80	15	0.10628E+03			0.10449E+03	0.16758E-01
90	15	0.12344E+03			0.12138E+03	0.16673E-01
100	15	0.14078E+03			0.13843E+03	0.16759E-01
110	15	0.15836E+03			0.15563E+03	0.17245E-01
21	20				0.92924E+01	
25	20				0.13897E+02	
30	20				0.20297E+02	
40	20				0.34375E+02	
50	20				0.49453E+02	
60	20				0.65133E+02	
70	20				0.81234E+02	
80	20				0.97635E+02	
90	20				0.11427E+03	
100	20				0.13111E+03	
110	20				0.14811E+03	
26	25				0.11672E+02	
30	25				0.16350E+02	
40	25				0.29602E+02	
50	25				0.44149E+02	
60	25				0.59451E+02	
70	25				0.75240E+02	
80	25				0.91391E+02	
90	25				0.10783E+03	
100	25				0.12445E+03	
110	25				0.14127E+03	

Table 6: *Square of the distance to suspects (see Section 5.1.2).*

k	d	L	N	M	M_1	M_2	K
30	$4, 5, \ldots, 15$	d	$k \cdot 2^L$	10^5	100	1000	8000
30	$16, 17, 18, 19$	d	$k \cdot 2^L$	-	-	-	8000
30	$20, 25, \ldots, 100$	d	$k \cdot 2^L$	-	-	-	8000

Table 7: *Parameters for Table 8 (see Section 5.1.2).*

d	L	$\mathbb{E}_{smpl}\left[D^{susp}\right]$	$\mathbb{E}_{smpl}\left[D_{algo}\right]$	err_{algo}	D^{susp}_{num}	err_{num}
4	4	0.15469E+01	0.15563E+01	0.41439E-02	0.15403E+01	0.42265E-02
5	5	0.19024E+01	0.19186E+01	0.36239E-02	0.19027E+01	0.10663E-03
6	6	0.22896E+01	0.23011E+01	0.30859E-02	0.22751E+01	0.63230E-02
7	7	0.26745E+01	0.26880E+01	0.29247E-02	0.26573E+01	0.63966E-02
8	8	0.30703E+01	0.30898E+01	0.26700E-02	0.30482E+01	0.71813E-02
9	9	0.34747E+01	0.34873E+01	0.25000E-02	0.34472E+01	0.78994E-02
10	10	0.38786E+01	0.39063E+01	0.23009E-02	0.38525E+01	0.67305E-02
11	11	0.43011E+01	0.43329E+01	0.23072E-02	0.42664E+01	0.80571E-02
12	12	0.47213E+01	0.47581E+01	0.21612E-02	0.46867E+01	0.73455E-02
13	13	0.51647E+01	0.51931E+01	0.19941E-02	0.51142E+01	0.97913E-02
14	14	0.56103E+01	0.56296E+01	0.19248E-02	0.55456E+01	0.11545E-01
15	15	0.60558E+01	0.60758E+01	0.18531E-02	0.59824E+01	0.12114E-01
16	16				0.64233E+01	
17	17				0.68714E+01	
18	18				0.73228E+01	
19	19				0.77782E+01	
20	20				0.82386E+01	
25	25				0.10585E+02	
30	30				0.13010E+02	
35	35				0.15492E+02	
40	40				0.18019E+02	
45	45				0.20598E+02	
50	50				0.23206E+02	
55	55				0.25856E+02	
60	60				0.28532E+02	
65	65				0.31232E+02	
70	70				0.33955E+02	
75	75				0.36700E+02	
80	80				0.39462E+02	
85	85				0.42250E+02	
90	90				0.45049E+02	
95	95				0.47865E+02	
100	100				0.50697E+02	

Table 8: *Square of the distance to suspects (see Section 5.1.2).*

and the sample variance

$$\mathrm{Var}_{smpl}\left[\mathrm{prop}_{algo}\right] = \frac{1}{M_2}\sum_{i=1}^{M_2}\left(\mathrm{prop}_{algo}(i) - \mathbb{E}_{smpl}\left[\mathrm{prop}_{algo}\right]\right)^2. \tag{209}$$

Also, we define the statistical error by the formula

$$err_{algo} = \frac{2}{\mathbb{E}_{smpl}\left[\mathrm{prop}_{algo}\right]} \cdot \sqrt{\frac{\mathrm{Var}_{smpl}\left[\mathrm{prop}_{algo}\right]}{M_2}}, \tag{210}$$

in complete analogy to (197), (198) and (199). The values (208), (210) appear in the third and fourth columns of Tables 10, 12, respectively.

Second way to approximate $\mathbb{E}\left[\mathbf{prop}_N\right]$: Theorem 16. This computation is completely analogous to the third way to compute $\mathbb{E}\left[D_N^{susp}\right]$ in Experiment 2 (Section 5.1.2). More specifically, we choose the positive integer parameter $K > 0$ and evaluate numerically the quantity P_{appr}, defined via (184). This evaluation consists of the following steps.

1. Choose equispaced discretization points $t_1, \ldots, t_{50}$ of the interval

$$I = \left(10^{-3}, 1 - 10^{-3}\right). \tag{211}$$

2. Define $\lambda_1, \ldots, \lambda_{50}$ by the formula

$$\lambda_i = F_{\chi_d^2}^{-1}(t_i), \tag{212}$$

for all $i = 1, \ldots, 50$, where $F_{\chi_d^2}^{-1}$ is the inverse of the cdf of χ_d^2 distribution (see (23)).

3. For each $i = 1, \ldots, 50$, generate K i.i.d. random vectors $\boldsymbol{a}_1^i, \ldots, \boldsymbol{a}_K^i$, having uniform distribution on the set $S_d^+(\sqrt{\lambda_i})$, defined via (11).

4. For each $i = 1, \ldots, 50$ and $j = 1, \ldots, K$, evaluate $P(\boldsymbol{a}_j^i)$, defined via (179). The evaluation is based on Theorem 9. More specifically, $G_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$ in (162) is computed via inverse Fourier transform of $h_{\boldsymbol{a}}^{\boldsymbol{\sigma}}$ defined by (86).

5. For each $i = 1, \ldots, 50$, compute the empirical average

$$\mathrm{Avg}(\lambda_i) = \frac{1}{K}\sum_{j=1}^{K} P(\boldsymbol{a}_j^i). \tag{213}$$

Obviously,

$$\underset{S_d^+(\sqrt{\lambda_i})}{\mathrm{Avg}}\left(P(\boldsymbol{a})\right) = \mathrm{Avg}(\lambda_i) + O\left(\frac{1}{\sqrt{K}}\right). \tag{214}$$

6. Evaluate the integral (184) by using the trapezoidal rule with nodes (212) and function values (213), i.e. by the formula

$$\mathrm{prop}_{num} = (t_2 - t_1) \cdot \left(\frac{1}{2}\left(\mathrm{Avg}(\lambda_1) + \mathrm{Avg}(\lambda_{50})\right) + \sum_{i=2}^{49}\mathrm{Avg}(\lambda_i)\right). \tag{215}$$

The value (215) appears in the fifth column of Tables 10, 12.

Structure of Tables 10, 12. The results of Experiment 3 are shown in Tables 10, 12 below. The first two columns contain the dimensionality d and the integer parameter L. The third and fourth columns contain the sample mean $\mathbb{E}_{smpl}\left[\text{prop}_{algo}\right]$ and its relative statistical error err_{algo}, defined via (208), (210), respectively. The sixth column contains prop_{num}, defined via (215). The last column contains the relative error err_{num} of approximating $\mathbb{E}_{smpl}\left[\text{prop}_{algo}\right]$ by prop_{num}, i.e.

$$err_{num} = \frac{\left|\text{prop}_{num} - \mathbb{E}_{smpl}\left[\text{prop}_{algo}\right]\right|}{\mathbb{E}_{smpl}\left[\text{prop}_{algo}\right]}. \tag{216}$$

Table 9 contains the choice of parameters k, d, L, N, M_1, M_2, K, corresponding to Table 10. Table 11 contains the choice of the same parameters, corresponding to Table 12.

k	d	L	N	M_1	M_2	K
30	$11, 20, 30, \ldots, 110$	10	30,720	100	1000	8000
30	$16, 20, 30, 40, 50$	15	983,040	100	100	8000
30	$60, 70, 80, 90, 100, 110$	15	983,040	-	-	8000
30	$21, 25, 30, 40, \ldots, 110$	20	31,457,280	-	-	8000
30	$26, 30, 40, \ldots, 110$	25	1,006,632,960	-	-	8000

Table 9: *Parameters for Table 10 (see Section 5.1.3).*

5.1.4 Description of Figures 2-5

In this section, we describe Figures 2-5, containing graphical representation of the data from Tables 2-12.

Figures 2(a), 2(b) correspond to Tables 4, 8. The parameters of the experiments are contained in Tables 3, 7, respectively.

In Figure 2(a), we compare $\mathbb{E}\left[D_N^{true}\right]$ (third column in Table 4) to D_{num}^{susp} (sixth column in Table 8). We plot these values, scaled by $d = L$, as functions of d. In Figure 2(b), we plot the ratio of D_{num}^{susp} to $\mathbb{E}\left[D_N^{true}\right]$, i.e.

$$\text{Ratio}_{true}^{susp} = \frac{D_{num}^{susp}}{\mathbb{E}\left[D_N^{true}\right]}. \tag{217}$$

Figures 3(a), 3(b), 4(a) correspond to Tables 2, 6. The parameters of the experiments are contained in Tables 1, 5, respectively.

Figure 3(a) is analogous to Figure 2(a). In this figure, we plot $\mathbb{E}\left[D_N^{true}\right]$ (third column in Table 2), scaled by d, as a function of the dimensionality d, for $L = 10, 15, 20, 25$. Figure 3(a) is also analogous to Figure 2(a). In this figure, we plot D_{num}^{susp} (sixth column in Table 6), scaled by d, as a function of the dimensionality d, for $L = 10, 15, 20, 25$.

Figure 4(a) is analogous to Figure 2(b). In this figure, we plot the ratio $\text{Ratio}_{true}^{susp}$ of $\mathbb{E}\left[D_N^{true}\right]$ to D_{num}^{susp} (see (217)), as a function of the dimensionality d, for $k = 30$ and $L = 10, 15, 20, 25$. The number of points N is determined via (121), as usual.

d	L	$\mathbb{E}_{smpl}\left[\text{prop}_{algo}\right]$	err_{algo}	prop_{num}	err_{num}
11	10	0.35653E+00	0.24860E-02	0.35719E+00	0.18464E-02
20	10	0.17270E+00	0.32763E-02	0.17280E+00	0.57214E-03
30	10	0.11052E+00	0.40884E-02	0.11049E+00	0.25339E-03
40	10	0.83474E-01	0.44436E-02	0.83198E-01	0.32985E-02
50	10	0.68766E-01	0.45101E-02	0.68199E-01	0.82308E-02
60	10	0.59150E-01	0.48360E-02	0.58718E-01	0.73146E-02
70	10	0.52820E-01	0.50961E-02	0.52163E-01	0.12438E-01
80	10	0.47708E-01	0.54266E-02	0.47366E-01	0.71617E-02
90	10	0.44104E-01	0.60874E-02	0.43704E-01	0.90541E-02
100	10	0.41222E-01	0.58727E-02	0.40792E-01	0.10455E-01
110	10	0.38680E-01	0.61434E-02	0.38448E-01	0.60067E-02
16	15	0.16063E+00	0.11954E-01	0.16254E+00	0.11910E-01
20	15	0.10312E+00	0.14959E-01	0.10310E+00	0.12858E-03
30	15	0.46549E-01	0.19168E-01	0.46303E-01	0.52846E-02
40	15	0.27540E-01	0.20330E-01	0.27389E-01	0.54829E-02
50	15	0.18663E-01	0.29931E-01	0.18818E-01	0.82879E-02
60	15			0.14139E-01	
70	15			0.11287E-01	
80	15			0.93643E-02	
90	15			0.80107E-02	
100	15			0.70247E-02	
110	15			0.62518E-02	
21	20			0.69683E-01	
25	20			0.42271E-01	
30	20			0.25180E-01	
40	20			0.11499E-01	
50	20			0.64951E-02	
60	20			0.41978E-02	
70	20			0.29569E-02	
80	20			0.22183E-02	
90	20			0.17408E-02	
100	20			0.14155E-02	
110	20			0.11795E-02	
26	25			0.28837E-01	
30	25			0.16947E-01	
40	25			0.59496E-02	
50	25			0.27427E-02	
60	25			0.14992E-02	
70	25			0.92644E-03	
80	25			0.62111E-03	
90	25			0.44054E-03	
100	25			0.33165E-03	
110	25			0.25772E-03	

Table 10: *Proportion of suspects among true nearest neighbors (see Section 5.1.3).*

k	d	L	N	M_1	M_2	K
30	$4, 5, \dots, 15$	d	$k \cdot 2^L$	100	1000	8000
30	$16, 17, 18, 19$	d	$k \cdot 2^L$	-	-	8000
30	$20, 25, \dots, 100$	d	$k \cdot 2^L$	-	-	8000

Table 11: *Parameters for Table 12 (see Section 5.1.3).*

d	L	$\mathbb{E}_{smpl}\left[\text{prop}_{algo}\right]$	err_{algo}	prop_{num}	err_{num}
4	4	0.84395E+00	0.10289E-02	0.83841E+00	0.65659E-02
5	5	0.76369E+00	0.12647E-02	0.76002E+00	0.48077E-02
6	6	0.68104E+00	0.14977E-02	0.68028E+00	0.11036E-02
7	7	0.60229E+00	0.17059E-02	0.60257E+00	0.45487E-03
8	8	0.52897E+00	0.20661E-02	0.52937E+00	0.74107E-03
9	9	0.46121E+00	0.22241E-02	0.46188E+00	0.14373E-02
10	10	0.39873E+00	0.24370E-02	0.40113E+00	0.60172E-02
11	11	0.34357E+00	0.27155E-02	0.34627E+00	0.78354E-02
12	12	0.29610E+00	0.29615E-02	0.29767E+00	0.52841E-02
13	13	0.25350E+00	0.31033E-02	0.25470E+00	0.47531E-02
14	14	0.21562E+00	0.35412E-02	0.21745E+00	0.84891E-02
15	15	0.18336E+00	0.37995E-02	0.18525E+00	0.10350E-01
16	16			0.15754E+00	
17	17			0.13331E+00	
18	18			0.11261E+00	
19	19			0.95102E-01	
20	20			0.80107E-01	
25	25			0.33351E-01	
30	30			0.13489E-01	
35	35			0.53638E-02	
40	40			0.21099E-02	
45	45			0.81603E-03	
50	50			0.31419E-03	
55	55			0.11943E-03	
60	60			0.45272E-04	
65	65			0.17071E-04	
70	70			0.64161E-05	
75	75			0.23932E-05	
80	80			0.89305E-06	
85	85			0.33126E-06	
90	90			0.12178E-06	
95	95			0.45417E-07	
100	100			0.16693E-07	

Table 12: *Proportion of suspects among true nearest neighbors (see Section 5.1.3).*

Figure 4(b) corresponds to Table 10. The parameters of the experiments are contained in Table 9.

In Figure 4(b), we plot prop_{num} (fifth column of Table 10). As demonstrated by Experiment 3 (see Section 5.1.3 above and Section 5.1.5 below), this quantity estimates the average proportion of the suspects among the true nearest neighbors (found by one iteration of RANN without supercharging). We plot prop_{num} as a function of the dimensionality d on the logarithmic scales, for fixed number of nearest neighbors $k = 30$ and $L = 10, 15, 20, 25$ (the number of points N is defined via (121), as usual).

In Figure 5, we plot prop_{num} (fifth column of Table 12) as a function of dimensionality d. Here the number of points N grows exponentially with dimensionality, via $d = L$ and (121). In addition, we illustrate the effects of degree of contact in the definition of V_i and $V_{\boldsymbol{\sigma}}$, defined via (101), (107), respectively. In other words, we demonstrate how prop_{num} will change if the algorithm searches for suspects of a point $\boldsymbol{x}_i \in B_{\boldsymbol{\sigma}}$ (see (100)) in the same box $B_{\boldsymbol{\sigma}}$ only (degree of contact zero, $V_{\boldsymbol{\sigma}}$ contains roughly k points), or in the boxes $B_{\boldsymbol{\mu}}$ having degree of contact up to two with $B_{\boldsymbol{\sigma}}$ (roughly $k \cdot L^2$ points).

Table 18 corresponds to Figure 5. Note that the first, second and fourth columns of Table 18 are identical to the first, second and fifth columns of Table 12, respectively (i.e. contain exactly the same data).

5.1.5 Observations

Several observations can be made from Experiments 1,2,3 (see Tables 2-12) and from the more detailed experiments by the authors. Some of these observations are illustrated by Figures 2-5, described in Section 5.1.4 above.

1. $\mathbb{E}\left[D_N^{true}\right]$ always overestimates $\mathbb{E}\left[D_{appr}^{true}\right]$; nevertheless, the relative error decreases with d and with L, and never exceeds 1% (see Tables 2, 4). Also, the quantities $\mathbb{E}\left[D_N^{true}\right]$ and $\mathbb{E}_{smpl}\left[D^{true}\right]$ (whenever the latter is available) coincide up to the precision of the calculation, as expected. Roughly speaking, this observation seems to indicate that the three way to compute $\mathbb{E}\left[D_N^{true}\right]$ (see Section 5.1.1) are equivalent, if low precision is required.

2. The values $\mathbb{E}_{smpl}\left[D^{susp}\right]$ and $\mathbb{E}_{smpl}\left[D_{algo}\right]$ coincide up to the precision of the calculation (roughly two significant decimal digits), as expected (see Tables 6, 8). In this case, we used the fact that the quantity $\mathbb{E}_{smpl}\left[D^{susp}\right]$ has been computed with relative error at most 0.5% (not shown in the tables). Roughly speaking, this seems to indicate that the first two ways to compute $\mathbb{E}\left[D_N^{susp}\right]$ (see Section 5.1.2) are equivalent, if low precision is required.

3. The quantity D_{num}^{susp} agrees with $\mathbb{E}_{smpl}\left[D^{susp}\right]$ up to the relative error below 2% (see Tables 6, 8). Additional numerical experiments demonstrate that the quantity D_{num}^{susp} has been computed with relative error at most 1% (not shown in the tables). This observation both validates Theorem 14 in Section 4.5.3 and seems to indicate that D_{num}^{susp} can be used to predict $\mathbb{E}_{smpl}\left[D^{susp}\right]$ up to roughly two decimal digits, when the latter is unavailable.

4. The quantities prop_{num} and $\mathbb{E}_{smpl}\left[\text{prop}_{algo}\right]$ agree roughly to two decimal digits, since err_{num} is below 2% (see Tables 10, 12). This observation seems to indicate that prop_{num} can be used to predict $\mathbb{E}_{smpl}\left[\text{prop}_{algo}\right]$ up to about two decimal digits, when the latter is unavailable.

5. The ratio of $\mathbb{E}\left[D_N^{true}\right]$ to the dimensionality d decreases with d and approaches the value about 0.25 (see Figure 2(a)). On the other hand, the ratio of D_{num}^{susp} to d seems to grow with d, reaching the value about 0.5 at $d = 100$ (see Figure 2(a)). Note that in Figures 2(a), 2(b) the dimensionality d roughly equals to the logarithm of the number of points N, due to (121).

6. The ratio $\text{Ratio}_{true}^{susp}$, defined via (217), slowly grows with d (see Figure 2(b)). This observation seems to indicate that the performance of the algorithm will deteriorate, if we keep the number of required nearest neighbors k fixed and let N grow exponentially. On the other hand, for $k = 30$, $d = 20$ and $N \approx 10^6$, even one iteration of the algorithm will result in $\text{Ratio}_{true}^{susp}$ being only about 1.4. In other words, the distance to the suspects will not be much larger than the distance to the true nearest neighbors.

7. The ratios of the square of the distances (to both true nearest neighbors and suspects) grow with dimensionality d, for a fixed number of nearest neighbors $k = 30$ and number of points N (see Figures 3(a), 3(b)). Also, for fixed k and d these quantities decrease with the number of points. For example, for $d = 40$, the value of $\mathbb{E}\left[D_N^{true}\right]$ changes from roughly d at $L = 10$ to roughly $d/2$ at $L = 25$.

8. The ratio $\text{Ratio}_{true}^{susp}$, defined via (217), slowly increases with the number of points N, for fixed values of number k of nearest neighbors and dimensionality d (see Figure 4(a)). In other words, the performance of RANN deteriorates with the number of points, if terms of $\text{Ratio}_{true}^{susp}$. However, as number of points is multiplied by 2, this ratio grows by a factor less than 1.1 for most values of d on Figure 4(a).

9. Perhaps surprisingly, the ratio $\text{Ratio}_{true}^{susp}$ decreases with dimensionality d, for fixed k and N (see Figure 4(a)). In other words, in terms of $\text{Ratio}_{true}^{susp}$, the performance of the algorithm actually slowly improves as dimensionality grows. For example, for $L = 15$ (and $N \approx 10^6$), this ratio decays from the value of about 1.57 at $d = 20$ to roughly 1.3 at $d = 110$.

10. The proportion prop_{num}, defined via (215), decays with d for fixed k, N, as expected (see Figure 4(b)). However, even for $d = 40$ and $N \approx 10^6$, RANN correctly finds about 2.7% of true nearest neighbors, on merely one iteration without supercharging. In other words, after 50 iterations without supercharging RANN will correctly detect about

$$75\% = 0.75 \approx 1 - (1 - 0.027)^{50} \tag{218}$$

true nearest neighbors, for $d = 40$ and $N \approx 10^6$ (see also Section 5.2.1).

11. The use of boxes with degree of contact up to two does not dramatically increase the proportion, while increasing the cost of a single iteration by a factor of d (see Figure 5).

In other words, L iterations of RANN, with $V_{\boldsymbol{\sigma}}$ defined via (107), will perform better than a single iteration of the version of RANN, that used boxes of degree of contact up to two. On the other hand, using only boxes of degree of contact zero will result in too few candidates for suspects. These observations seem to indicate that the value 1 of degree of contact in (101), (107) strikes the right balance, in terms of cost vs. performance. Additional numerical experiments, whose results are not reported in this paper, seem to confirm this statement.

5.2 Illustration of the performance of the algorithm

In this section, we illustrate the performance of RANN (described in Sections 4.2, 4.3) via several numerical examples. In particular, we demonstrate how the performance is affected by the initial distribution of the points and supercharging (see Sections 4.2.2, 4.3.3).

5.2.1 Experiment 4: performance of RANN

In this experiment, we run RANN with various parameters on different sets of points and report on the resulting statistics.

Description of the experiment. We choose the dimensionality d and the number of points N. Then we choose the number of nearest neighbors k, the number of iterations of the algorithm $T > 0$, and whether to perform the supercharging or not. Also, we choose positive integer parameters $1 \leq M_1 < N$ and $M_2 > 0$. Next, we generate N vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$ in $\mathbb{R}^d$ in one of the following three different ways:

1. $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$ are independent standard normal random vectors in $\mathbb{R}^d$.

2. $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$ are independent random vectors in $\mathbb{R}^d$, distributed uniformly in the hypercube $[0,1]^d$. In other words, to generate each $\boldsymbol{x}_i$ we generate d independent random variables $U_i(1), \dots, U_i(d)$, distributed uniformly in $[0,1]$, and set
$$\boldsymbol{x}_i = (U_i(1), \dots, U_i(d))\,. \tag{219}$$

3. $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$ are independent vectors in $\mathbb{R}^d$, distributed uniformly on the vertices of the Hamming cube $\{0,1\}^d$. More specifically, to generate each $\boldsymbol{x}_i$ we generate d independent random variables $V_i(1), \dots, V_i(d)$, taking values 0 or 1 with probability 1/2. Then, we define $\boldsymbol{x}_i$ by the formula
$$\boldsymbol{x}_i = (V_i(1), \dots, V_i(d))\,. \tag{220}$$

   In other words, each coordinate of any $\boldsymbol{x}_i$ is either zero or one, with equal probability.

We implement RANN and run it on $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$. For each $\boldsymbol{x}_i$, RANN finds its k suspects, $\boldsymbol{x}_{s(i,1)}, \dots, \boldsymbol{x}_{s(i,k)}$. Also, for all $i = 1, \dots, M_1 < N$, we find the list $\boldsymbol{x}_{t(i,1)}, \dots, \boldsymbol{x}_{t(i,k)}$ of k true nearest neighbors of $\boldsymbol{x}_i$, by direct scanning. Then, we compute the quantities D_i^{true}, D_i^{susp}, prop_i, defined via (118), (119), (120), respectively, for $i = 1, \dots, M_1$. We define the average D_{algo}^{true} by the formula
$$D_{algo}^{true} = \frac{1}{M_1} \sum_{i=1}^{M_1} D_i^{true}, \tag{221}$$

and the averages D_{algo}^{susp}, prop_{algo} via (196), (207), respectively.

Generation of the points and invocation of RANN are repeated M_2 times, to obtain the values $D_{algo}^{true}(1)$, ..., $D_{algo}^{true}(M_2)$, all defined via (221). Then, we define the sample mean of D_{algo}^{true} by the formula

$$\mathbb{E}_{smpl}\left[D^{true}\right] = \frac{1}{M_2} \sum_{i=1}^{M_2} D_{algo}^{true}(i). \tag{222}$$

The sample means $\mathbb{E}_{smpl}\left[D_{algo}\right]$ and $\mathbb{E}_{smpl}\left[\text{prop}_{algo}\right]$ are defined via (197), (208), respectively. Finally, we define the ratio of $\mathbb{E}_{smpl}\left[D_{algo}\right]$ to $\mathbb{E}_{smpl}\left[D^{true}\right]$ by the formula

$$\text{ratio}_{smpl} = \frac{\mathbb{E}_{smpl}\left[D_{algo}\right]}{\mathbb{E}_{smpl}\left[D^{true}\right]}. \tag{223}$$

Structure of Tables 14, 15, 16. Initially, Experiment 4 was conducted 24 times, with all the combinations of values of the parameters from Table 13. These parameters were chosen to demonstrate the effects of supercharging on a particular case (fixed dimensionality d and number of points N, Gaussian distribution of the points). The results are shown in Tables 14, 15, 16. See also Figures 6, 7, 8.

Table 14 has the following structure. The first column contains the number k of required nearest neighbors. The next three columns contain $\text{ratio}_{smpl} - 1$, where ratio_{smpl} is defined via (223), for $1, 5, 10$ iterations of RANN, respectively. To obtain these values, the supercharging step was skipped. The last three columns contain the same value, for $1, 5, 10$ iterations of RANN, respectively, followed by supercharging.

Table 15 has the same structure as Table 14, but instead of $\text{ratio}_{smpl} - 1$ it contains the value $\mathbb{E}_{smpl}\left[\text{prop}_{algo}\right]$, defined via (208). The values in both Tables 14, 15 have relative error less than 1%.

Table 16 has the same structure as Tables 14, 15, but contains the running time of RANN in seconds. We recall that the algorithm was compiled and run on a modern laptop computer, with Dual Core 2.53 GHz CPU and 2.9 Gb RAM.

Structure of Tables 19 - 45. Next, Experiment 4 was conducted 1008 times, for all the combinations of values the parameters from Table 17. In particular, different values of dimensionality d and different initial distributions of points were used. The results are shown in Tables 19 - 45 and also displayed in the corresponding Figures 6 - 23.

The structure of Tables 19 - 45 is similar, but not identical, to that of Tables 14, 15, 16. The first column contains the dimensionality d. The next four columns correspond to different parameters of RANN (number of iterations T, with or without supercharging). The second and third columns correspond to RANN without supercharging, $T = 1, 10$, respectively. The fourth and fifth columns correspond to RANN with supercharging, $T = 1, 10$, respectively.

The value shown in Columns 2-5 is one of the three statistics, illustrating the performance of RANN. In Tables 19, 21, 23, 25, 27, 29, 31, 33, 35, we show ratio_{smpl}, defined via (223). In Tables 20, 22, 24, 26, 28, 30, 32, 34, 36, we show $\mathbb{E}_{smpl}\left[\text{prop}_{algo}\right]$, defined via (208). In Tables 37 - 45, we show the running time of RANN in seconds.

Also, Tables 19 - 45 differ by the requested number of nearest neighbors k (=15, 30, 60), and the distribution of the points (normal, uniform, Hamming cube). Most of the results have been computed with relative error about 2%.

Description of Figures 6 - 23. Figures 6 - 23 correspond to Tables 19 - 45. On these figures, the corresponding quantity (ratio_{smpl}, $\mathbb{E}_{smpl}\left[\text{prop}_{algo}\right]$ or running time) is plotted as a function of the dimensionality d. Each figure contains four curves, corresponding to one iteration of RANN without supercharging, ten iterations of RANN without supercharging, one iteration of RANN with supercharging and ten iterations of RANN with supercharging.

5.2.2 Observations

Several observations can be made from Tables 14, 15, 16.

1. For a fixed k, the proportion of suspects among the true nearest neighbors grows exponentially with number of iterations T (without supercharging), as expected. For example, for $k = 15$, one iteration of RANN determines about 6.2% of true nearest neighbors correctly (second row, third column in Table 15). After 5 iterations, the proportion is

$$26.9\% = 0.269 \approx \left(1 - (1 - 0.062)^5\right). \tag{224}$$

After 10 iterations, the proportion is

$$46.3\% = 0.463 \approx \left(1 - (1 - 0.062)^{10}\right). \tag{225}$$

2. Supercharging significantly improves the performance of the algorithm. For example, for $k = 120$, the proportion of correctly determined true nearest neighbors is about 68%, after 10 iterations of RANN without supercharging. If we also perform supercharging, this proportion grows to 99.2%.

3. The ratio (223) decreases with number T of iterations, and also if supercharging is performed (see Table 14).

4. The performance of the algorithm improves as the number of nearest neighbors k grows (for fixed number of points $N = 122,880$ and dimensionality $d = 30$). This accuracy comes at the expense of running time of the algorithm (see Section 4.4). We recall that running time grows roughly linearly with k if no supercharging is done, and the cost of supercharging grows quadratically with k (see (114)). Table 16 seems to confirm the predicted cost of the algorithm, as expected.

Also, several observations can be done from Tables 19 - 45 and Figures 6 - 23. In these observations, we refer to ratio_{smpl} as "ratio", and to $\mathbb{E}_{smpl}\left[\text{prop}_{algo}\right]$ as "proportion". We recall that, roughly speaking, the proportion measures how many of the true nearest neighbors have been found by RANN. On the other hand, the ratio measures how much average distances to suspects differ from the average distances to true nearest neighbors.

1. As expected, for a fixed d the performance of RANN improves as the number T of iterations increases, both in terms of ratio and proportion.

parameter	values
d	30
N	$122880 = 30 \cdot 2^{12}$
M_1	1000
M_2	10
k	15, 30, 60, 120
T	1, 5, 10
perform supercharging	yes, no

Table 13: *Parameters for Tables 14, 15, 16 (see Section 5.2.1).*

	without supercharging			with supercharging		
k	$T = 1$	$T = 5$	$T = 10$	$T = 1$	$T = 5$	$T = 10$
15	0.355E+00	0.128E+00	0.650E-01	0.239E+00	0.724E-01	0.350E-01
30	0.328E+00	0.110E+00	0.521E-01	0.200E+00	0.356E-01	0.143E-01
60	0.302E+00	0.929E-01	0.406E-01	0.177E+00	0.112E-01	0.334E-02
120	0.275E+00	0.764E-01	0.302E-01	0.160E+00	0.186E-02	0.406E-03

Table 14: *(Ratio-1) with and without supercharging (see Section 5.2.1).*

	without supercharging			with supercharging		
k	$T = 1$	$T = 5$	$T = 10$	$T = 1$	$T = 5$	$T = 10$
15	0.622E-01	0.269E+00	0.463E+00	0.147E+00	0.438E+00	0.630E+00
30	0.756E-01	0.318E+00	0.531E+00	0.202E+00	0.636E+00	0.806E+00
60	0.919E-01	0.375E+00	0.604E+00	0.247E+00	0.848E+00	0.944E+00
120	0.112E+00	0.441E+00	0.681E+00	0.288E+00	0.969E+00	0.992E+00

Table 15: *Proportion with and without supercharging (see Section 5.2.1).*

	without supercharging			with supercharging		
k	$T = 1$	$T = 5$	$T = 10$	$T = 1$	$T = 5$	$T = 10$
15	0.227E+01	0.123E+02	0.245E+02	0.707E+01	0.176E+02	0.299E+02
30	0.386E+01	0.217E+02	0.443E+02	0.208E+02	0.408E+02	0.639E+02
60	0.715E+01	0.416E+02	0.841E+02	0.667E+02	0.112E+03	0.155E+03
120	0.140E+02	0.834E+02	0.176E+03	0.241E+03	0.364E+03	0.445E+03

Table 16: *Running times of RANN, in seconds (see Section 5.2.1).*

2. As expected, for a fixed d the performance of RANN improves if supercharging is performed, both in terms of ratio and proportion.

3. For a fixed d, the performance of RANN improves as the number of requested nearest neighbors k increases, both in terms of ratio and proportion (at the expense of running time).

4. For a fixed d, the effects of supercharging (especially on proportion) increase as k grows. For example, in Figure 6(b) (normal distribution, $k = 15$) we observe, that, for $T = 10$ and $d = 60$, supercharging increases the proportion from 22% to 32% (see the third and fifth columns in Table 20). On the other hand, in Figure 8(b) (normal distribution, $k = 60$) we observe that, for $T = 10$ and $d = 60$, supercharging increases the proportion from 36% to 77% (see the third and fifth columns in Table 24).

5. As expected, the performance of RANN slowly deteriorates in terms of proportion, as d increases. On the other hand, there is no significant deterioration of performance in terms of ratio, as d increases.

6. In all the tables and figures, for $T = 10$ the ratio is always below 1.1, with or without supercharging.

7. In all the tables and figures, even for as high a dimension as $d = 60$, as few as ten iterations with supercharging correctly determine at least 30% of the true nearest neighbors. Moreover, the error of this detection decays exponentially with number of iterations T (see also (224), (225)).

8. Tables 19 - 36 and Figures 6 - 14 seem to indicate that RANN is relatively insensitive to whether the initial distribution of the points is normal, uniform (in the $d-$dimensional hypercube) or Hamming (uniform on the vertices of the Hamming cube $\{0,1\}^d$).

9. Tables 37 - 45 and Figures 15 - 23 seem to indicate that the running time of RANN does not depend on the inital distribution of the points, as expected.

10. The running time of RANN, with or without supercharging, grows roughly linearly with dimensionality d, as expected from (114) (see Tables 37 - 45 and Figures 15 - 23).

11. For fixed dimensionality d, the running time of RANN without supercharging grows roughly linearly with the requested number of nearest neighbors k, as expected from (114) (see Tables 37 - 45 and Figures 15 - 23). For example, 10 iterations of RANN in the case of $d = 200$ take about 80, 124 and 208 seconds for $k = 15, 30, 60$, respectively (see the third column in Tables 39, 42, 45).

12. For fixed dimensionality d, the running time of the supercharging step grows roughly quadratically with the requested number of nearest neighbors k, as expected from (114) (see Tables 37 - 45 and Figures 15 - 23). For example, in the case of $d = 200$ supercharging takes about 15, 60 and 230 seconds for $k = 15, 30, 60$, respectively.

parameter	values
N	$122880 = 30 \cdot 2^{12}$
M_1	1000
M_2	5
d	15, 20, ..., 95, 100, 110, ..., 200
k	15, 30, 60
T	1,10
perform supercharging	yes, no
points distribution	normal, uniform, Hamming cube

Table 17: *Parameters for Tables 19 - 45 and Figures 6 - 23 (see Section 5.2.1).*

6 Miscellaneous

6.1 Version of RANN for highly asymmetric distributions

In Section 4.2.1, we describe the division of all the points B into boxes $B_{\boldsymbol{\sigma}}$ (see e.g. (96), (97), (98), (100)). Each box $B_{\boldsymbol{\sigma}}$ contains roughly k points, if the points in B are approximately radially symmetric. For a highly asymmetric distribution of points, this is no longer the case. This might result in a large proportion boxes $B_{\boldsymbol{\sigma}}$ having too few or too many points.

Fortunately, a slight alteration of the construction of the boxes eliminates this problem for asymmetric distributions of points. Namely, the division is done not by the origin, as in (100), but by the median of the corresponding coordinate. More specifically, we choose a real number $y(1)$ such that the first coordinate of half of the points in B is less than $y(1)$. In other words,

$$|\{\boldsymbol{x} \in B : x(1) < y(1)\}| = \lfloor N/2 \rfloor. \tag{226}$$

We divide all the points into two disjoint sets

$$B_- = \{\boldsymbol{x} \in B : x(1) < y(1)\}, \quad B_+ = \{\boldsymbol{x} \in B : x(1) \geq y(1)\}. \tag{227}$$

Obviously, the sizes of B_- and B_+ are the same if N is even and differ by one if N is odd. Next, we set $y_+(2)$ to be a real number such that the second coordinate of half of the points in B_+ is less than $y_+(2)$, i.e.

$$|\{\boldsymbol{x} \in B_+ : x(2) < y_+(2)\}| = \lfloor N/4 \rfloor. \tag{228}$$

We split B_+ into two disjoint sets B_{+-} and B_{++} by the same principle, e.g.

$$B_{+-} = \{\boldsymbol{x} \in B_+ : x(2) < y_+(2)\}, \quad B_{++} = \{\boldsymbol{x} \in B_+ : x(2) \geq y_+(2)\}. \tag{229}$$

We construct B_{--} and B_{-+} in a similar fashion by using a real number $y_-(2)$ such that the second coordinate of half of the points in B_- is less than $y_-(2)$, i.e.

$$|\{\boldsymbol{x} \in B_- : x(2) < y_-(2)\}| = \lfloor N/4 \rfloor. \tag{230}$$

Each of the four boxes $B_{--}, B_{-+}, B_{+-}, B_{++}$ contains either $\lfloor N/4 \rfloor$ or $\lfloor N/4 \rfloor + 1$ points. Then we repeat the subdivision by splitting each of the four boxes into two by using the third coordinate, and so on. We proceed until we end up with a collection of 2^L boxes $\{B_{\boldsymbol{\sigma}}\}$ with k or $k+1$ points in each box. Here the box index $\boldsymbol{\sigma}$ is a word of symbols $+, -$ of length L as in (3) and L is a positive integer such that $k \cdot 2^L \leq N < k \cdot 2^{L+1}$.

Extensive numerical experiments seem to indicate that for normally distributed points there are no significant differences in performance between this version of RANN and the one described in Section 4.2.1.

References

[1] M. Abramowitz, I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables*, Dover Publications (1964).

[2] N. Ailon, B. Chazelle, *The Fast Johnson–Lindenstrauss Transform and Approximate Nearest Neighbors*, SIAM J. Comput., 39(1):302-322, 2009.

[3] N. Ailon, B. Chazelle, *Faster Dimension Reduction*, Commun. ACM, 53(2):97-104, 2010.

[4] N. Ailon, E. Liberty, *Almost Optimal Unrestricted Fast Johnson-Lindenstrauss Transform*, eprint arXiv:1005.5513, 2010.

[5] Arya, S., D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, *An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions*, Journal of the ACM, vol. 45, no. 6, pp. 891-923 (1998).

[6] P. Billingsley, *Probability and Measure*, Wiley, NY (1986).

[7] W. Feller, *An Introduction to Probability Theory and Its Applications*, Volume II, Second edition, Wiley, NY (1957).

[8] G. R. Grimmett and D. R. Stirzaker, *Probability and Random Processes*, Second edition, Oxford University Press (1992).

[9] A. Andoni, P. Indyk *Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions*, Communications of the ACM, vol. 51, no. 1, 2008, p. 117-122.

[10] W. Johnson and J. Lindenstrauss, *Extensions of Lipschitz mappings into a Hilbert space*, Contemporary Mathematics, 26:189206, 1984.

[11] I. Katznelson, *An Introduction to Harmonic Analysis*, Second edition, Dover Publications (1976).

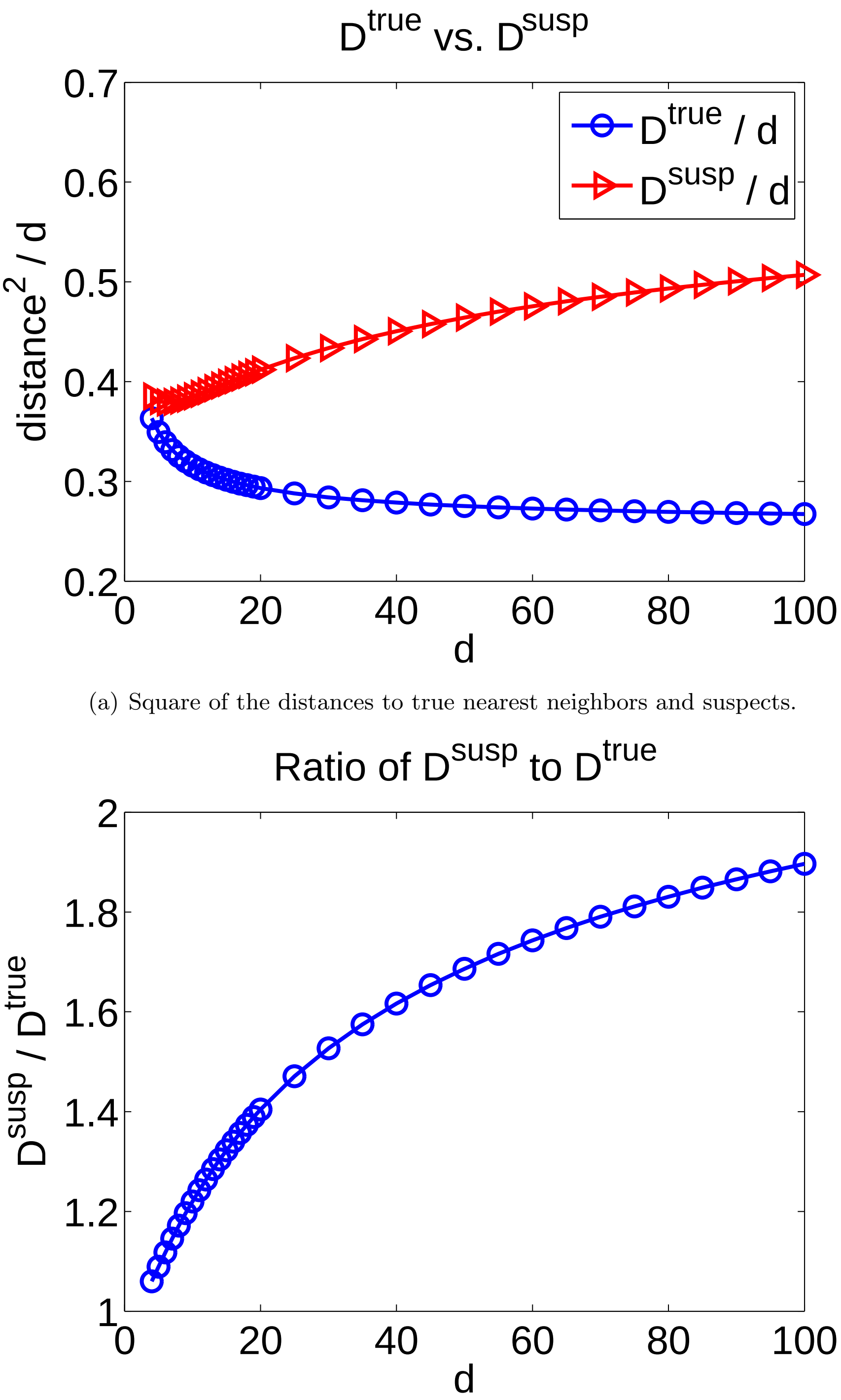
[12] D. Knuth, *Seminumerical Algorithms, vol. 2 of The Art of Computer Programming*, Reading, Mass: Addison-Wesley (1969).

[13] Vladimir Rokhlin and Mark Tygert, *A fast randomized algorithm for overdetermined linear least-squares regression*, Proceedings of the National Academy of Sciences USA, 105 (36): 13212-13217, 2008.

[14] Edo Liberty, Franco Woolfe, Per-Gunnar Martinsson, Vladimir Rokhlin, Mark Tygert, *Randomized algorithms for the low-rank approximation of matrices*, Proceedings of the National Academy of Sciences USA 104 : 20167 20172, 2007.

[15] W. Rudin, *Functional Analysis*, Mc-Graw Hill (1973).

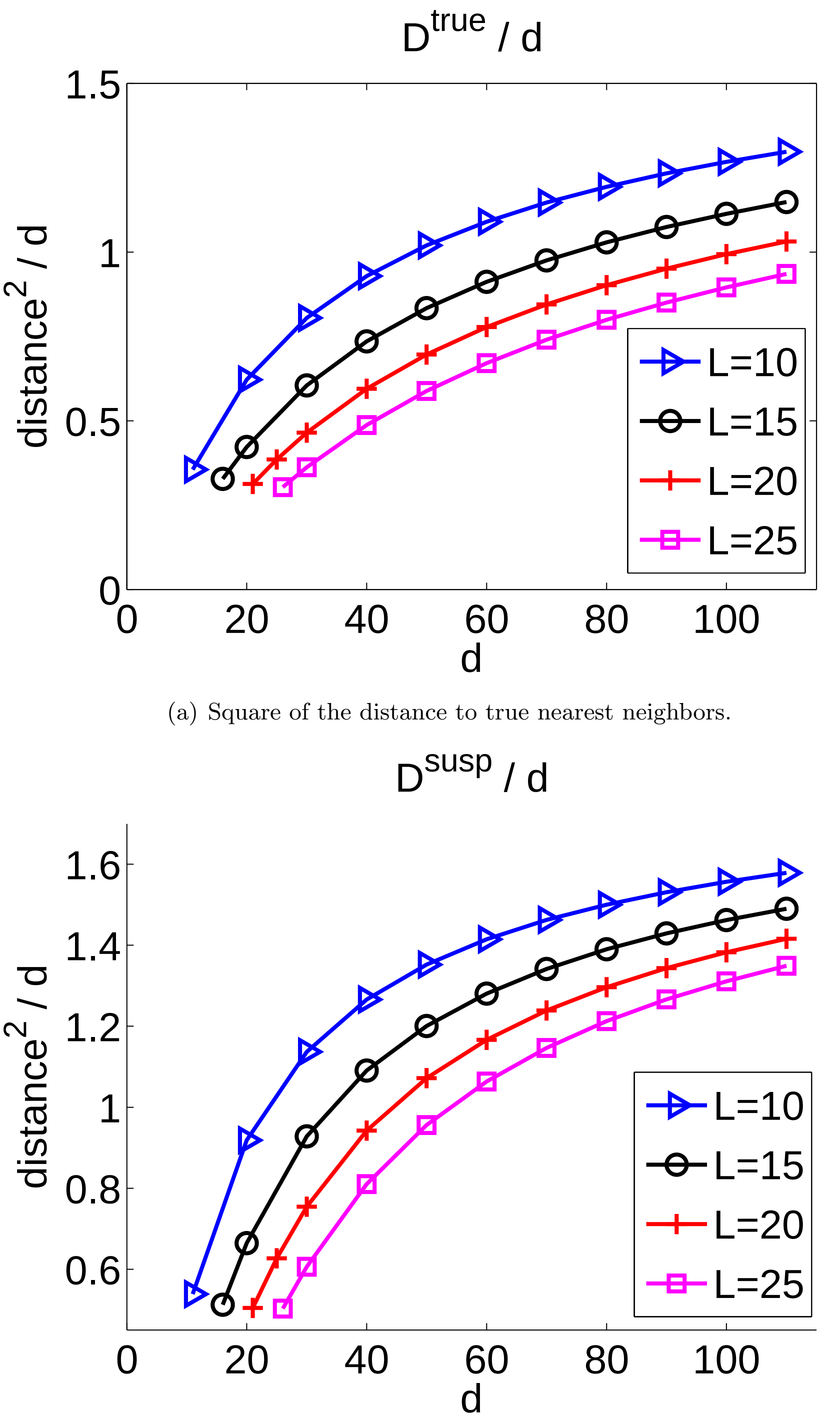
[16] W. Rudin, *Real and Complex Analysis*, Mc-Graw Hill (1970).



(a) Square of the distances to true nearest neighbors and suspects.

(b) Ratio of the average square of the distances, suspects vs. true nearest neighbors.

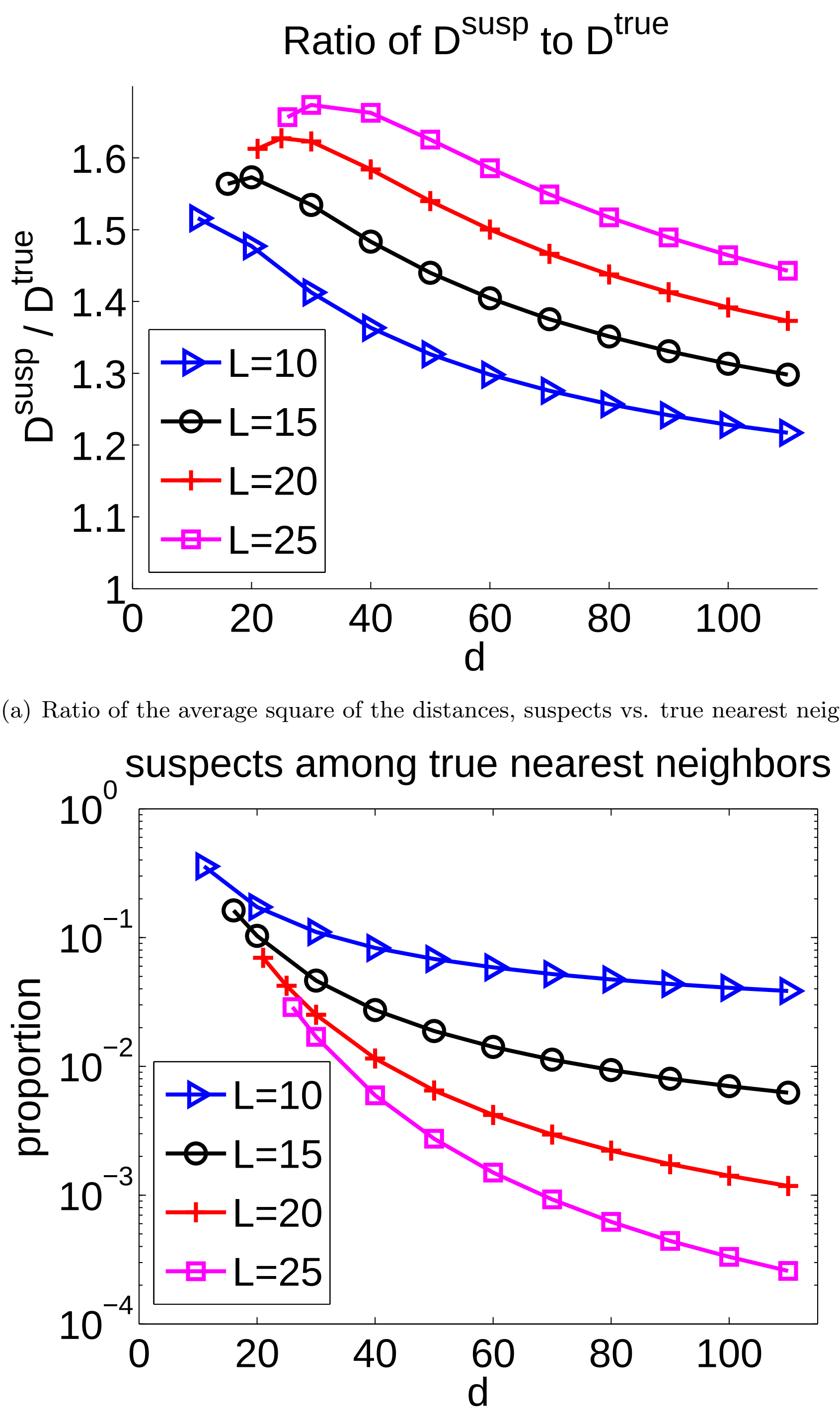
Figure 2: Suspects vs. true nearest neighbors (see Section 5.1.4). Number of points: $N = 30 \cdot 2^d$, where d is the dimensionality. Number of requested nearest neighbors: $k = 30$. See also Tables 4, 8.



(a) Square of the distance to true nearest neighbors.

(b) Square of the distance to suspects.

Figure 3: Distances to suspects and true nearest neighbors (see Section 5.1.4). Number of points: $N = 30 \cdot 2^L$, i.e. $L = \log_2(N/30)$. Dimensionality: d. Number of requested nearest neighbors: $k = 30$. See also Tables 2, 6.



(a) Ratio of the average square of the distances, suspects vs. true nearest neighbors.

(b) Proportion of suspects among true nearest neighbors.

Figure 4: Statistics of the suspects (see Section 5.1.4). Number of points: $N = 30 \cdot 2^L$, i.e. $L = \log_2 (N/30)$. Dimensionality: d. Number of requested nearest neighbors: $k = 30$. See also Tables 2, 6.

d	L	contact $= 0$	contact ≤ 1	contact ≤ 2
4	4	0.39707E+00	0.83841E+00	0.96819E+00
5	5	0.31606E+00	0.76002E+00	0.94832E+00
6	6	0.25239E+00	0.68028E+00	0.91829E+00
7	7	0.20182E+00	0.60257E+00	0.87922E+00
8	8	0.16169E+00	0.52937E+00	0.83286E+00
9	9	0.12968E+00	0.46188E+00	0.78127E+00
10	10	0.10428E+00	0.40113E+00	0.72680E+00
11	11	0.83833E-01	0.34627E+00	0.67008E+00
12	12	0.67415E-01	0.29767E+00	0.61341E+00
13	13	0.54180E-01	0.25470E+00	0.55745E+00
14	14	0.43616E-01	0.21745E+00	0.50388E+00
15	15	0.35173E-01	0.18525E+00	0.45313E+00
16	16	0.28403E-01	0.15754E+00	0.40564E+00
17	17	0.22867E-01	0.13331E+00	0.36088E+00
18	18	0.18419E-01	0.11261E+00	0.31984E+00
19	19	0.14873E-01	0.95102E-01	0.28260E+00
20	20	0.12000E-01	0.80107E-01	0.24868E+00
25	25	0.41318E-02	0.33351E-01	0.12582E+00
30	30	0.14250E-02	0.13489E-01	0.60007E-01
35	35	0.49412E-03	0.53638E-02	0.27496E-01
40	40	0.17248E-03	0.21099E-02	0.12242E-01
45	45	0.59931E-04	0.81603E-03	0.52915E-02
50	50	0.20958E-04	0.31419E-03	0.22509E-02
55	55	0.72935E-05	0.11943E-03	0.93752E-03
60	60	0.25508E-05	0.45272E-04	0.38613E-03
65	65	0.89258E-06	0.17071E-04	0.15728E-03
70	70	0.31314E-06	0.64161E-05	0.63468E-04
75	75	0.10943E-06	0.23932E-05	0.25316E-04
80	80	0.38426E-07	0.89305E-06	0.10056E-04
85	85	0.13456E-07	0.33126E-06	0.39567E-05
90	90	0.46833E-08	0.12178E-06	0.15389E-05
95	95	0.16605E-08	0.45417E-07	0.60439E-06
100	100	0.58117E-09	0.16693E-07	0.23357E-06

Table 18: Proportion of suspects among true nearest neighbors for different variations of RANN (see Section 5.1.4). Number of points: $N = 30 \cdot 2^d = 30 \cdot 2^L$, where d is the dimensionality. Number of requested nearest neighbors: $k = 30$. The last three columns correspond to different ways to select suspects of a point on an iteration of RANN (see Section 4.2.1). Suspects are selected from the same box (third column), from $d + 1$ boxes with degree of contact up to one (fourth column), from $d^2/2 + d/2 + 1$ boxes with degree of contact up to two (fifth column). Corresponds to Figure 5.

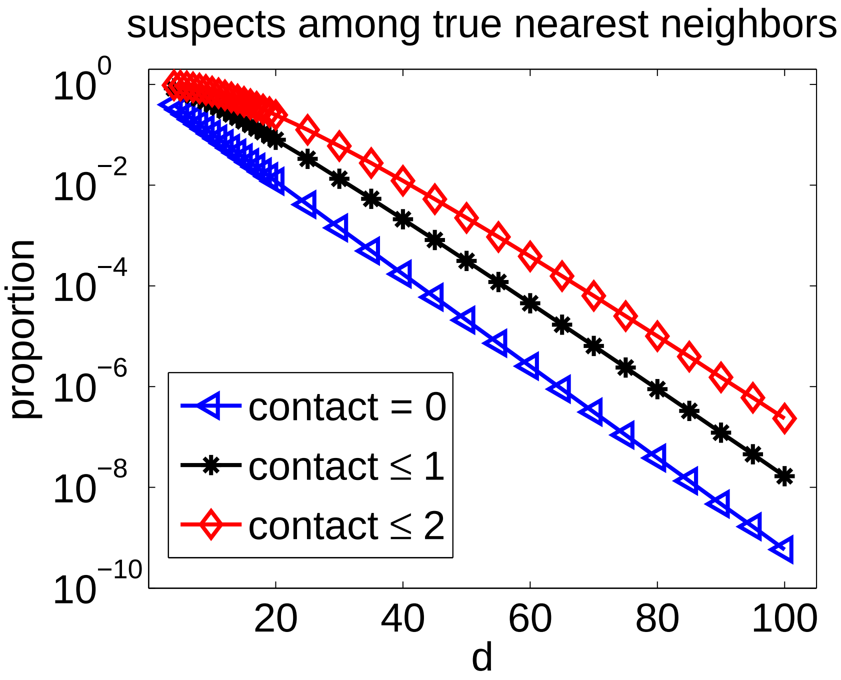


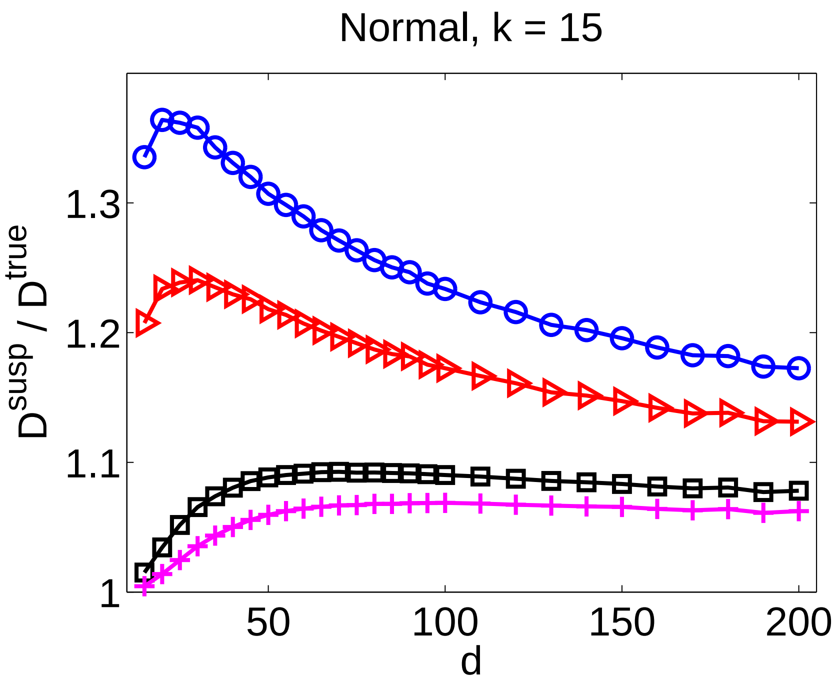
Figure 5: Proportion of suspects among true nearest neighbors for different variations of RANN (see Section 5.1.4). Number of points: $N = 30 \cdot 2^d$, where d is the dimensionality. Number of requested nearest neighbors: $k = 30$. The three curves correspond to different ways to select suspects of a point on an iteration of RANN (see Section 4.2.1). Suspects are selected from the same box (triangles), $d + 1$ boxes with degree of contact up to one (asterisks), $d^2/2 + d/2 + 1$ boxes with degree of contact up to two (diamonds). See Table 18.

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.13353E+01	0.10149E+01	0.12074E+01	0.10044E+01
20	0.13640E+01	0.10344E+01	0.12336E+01	0.10138E+01
25	0.13617E+01	0.10516E+01	0.12386E+01	0.10246E+01
30	0.13579E+01	0.10655E+01	0.12404E+01	0.10353E+01
35	0.13428E+01	0.10738E+01	0.12349E+01	0.10435E+01
40	0.13309E+01	0.10805E+01	0.12295E+01	0.10501E+01
45	0.13201E+01	0.10855E+01	0.12257E+01	0.10556E+01
50	0.13071E+01	0.10883E+01	0.12180E+01	0.10594E+01
55	0.12984E+01	0.10902E+01	0.12136E+01	0.10624E+01
60	0.12896E+01	0.10912E+01	0.12069E+01	0.10642E+01
65	0.12790E+01	0.10924E+01	0.12015E+01	0.10658E+01
70	0.12710E+01	0.10926E+01	0.11967E+01	0.10668E+01
75	0.12634E+01	0.10920E+01	0.11918E+01	0.10670E+01
80	0.12559E+01	0.10921E+01	0.11869E+01	0.10679E+01
85	0.12503E+01	0.10918E+01	0.11835E+01	0.10680E+01
90	0.12465E+01	0.10915E+01	0.11816E+01	0.10685E+01
95	0.12378E+01	0.10908E+01	0.11752E+01	0.10686E+01
100	0.12336E+01	0.10901E+01	0.11725E+01	0.10687E+01
110	0.12233E+01	0.10890E+01	0.11665E+01	0.10681E+01
120	0.12158E+01	0.10872E+01	0.11610E+01	0.10673E+01
130	0.12060E+01	0.10856E+01	0.11540E+01	0.10666E+01
140	0.12019E+01	0.10846E+01	0.11514E+01	0.10660E+01
150	0.11956E+01	0.10833E+01	0.11472E+01	0.10656E+01
160	0.11884E+01	0.10813E+01	0.11419E+01	0.10640E+01
170	0.11825E+01	0.10799E+01	0.11376E+01	0.10630E+01
180	0.11819E+01	0.10805E+01	0.11381E+01	0.10639E+01
190	0.11738E+01	0.10770E+01	0.11316E+01	0.10611E+01
200	0.11725E+01	0.10781E+01	0.11313E+01	0.10623E+01

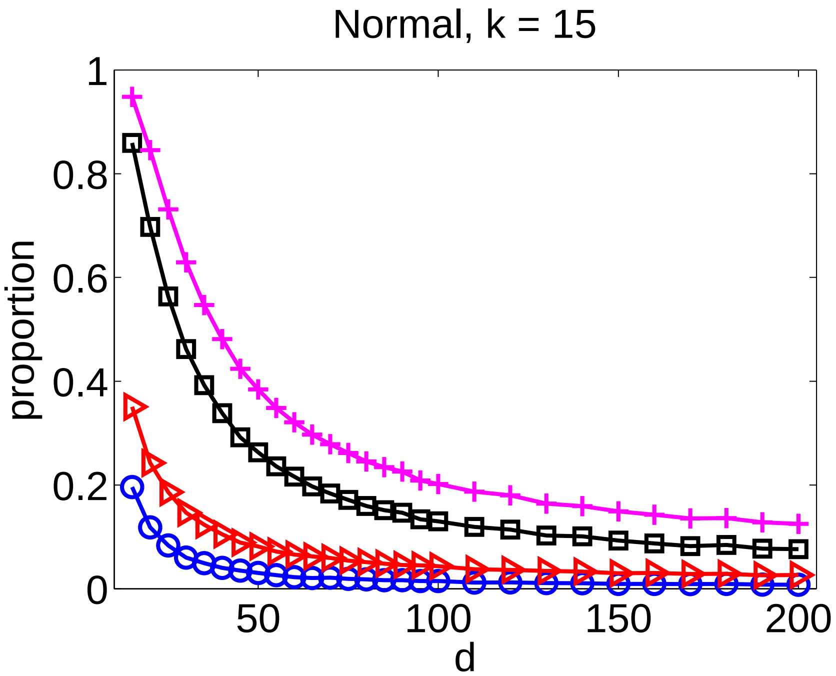
Table 19: Ratio of the average square of the distances, suspects vs. true nearest neighbors. Normal distribution, number of requested nearest neighbors: $k = 15$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 6(a).

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.19584E+00	0.85949E+00	0.35073E+00	0.94827E+00
20	0.11812E+00	0.69748E+00	0.24260E+00	0.84541E+00
25	0.83533E-01	0.56327E+00	0.18616E+00	0.73115E+00
30	0.59866E-01	0.46200E+00	0.14638E+00	0.62908E+00
35	0.49133E-01	0.39225E+00	0.12353E+00	0.54678E+00
40	0.40066E-01	0.33843E+00	0.10472E+00	0.48118E+00
45	0.34799E-01	0.29185E+00	0.89093E-01	0.42382E+00
50	0.30333E-01	0.26293E+00	0.81440E-01	0.38430E+00
55	0.26186E-01	0.23582E+00	0.71559E-01	0.34853E+00
60	0.22360E-01	0.21615E+00	0.65813E-01	0.32086E+00
65	0.20720E-01	0.19719E+00	0.62333E-01	0.29700E+00
70	0.21173E-01	0.18346E+00	0.59292E-01	0.27857E+00
75	0.19026E-01	0.17113E+00	0.53986E-01	0.26147E+00
80	0.17946E-01	0.15935E+00	0.51159E-01	0.24525E+00
85	0.16346E-01	0.15135E+00	0.48399E-01	0.23414E+00
90	0.16346E-01	0.14660E+00	0.45799E-01	0.22597E+00
95	0.14813E-01	0.13374E+00	0.45200E-01	0.20868E+00
100	0.14866E-01	0.12980E+00	0.43453E-01	0.20171E+00
110	0.11746E-01	0.11934E+00	0.37506E-01	0.18730E+00
120	0.12066E-01	0.11392E+00	0.36106E-01	0.17990E+00
130	0.11000E-01	0.10250E+00	0.34146E-01	0.16421E+00
140	0.10493E-01	0.10093E+00	0.32853E-01	0.15930E+00
150	0.92799E-02	0.92973E-01	0.29879E-01	0.14907E+00
160	0.91333E-02	0.87453E-01	0.30413E-01	0.14243E+00
170	0.89599E-02	0.81959E-01	0.28520E-01	0.13538E+00
180	0.91199E-02	0.84386E-01	0.28719E-01	0.13603E+00
190	0.78266E-02	0.77360E-01	0.25759E-01	0.12801E+00
200	0.77333E-02	0.76119E-01	0.26453E-01	0.12505E+00

Table 20: Proportion of suspects among true nearest neighbors. Normal distribution, number of requested nearest neighbors: $k = 15$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 6(b).



(a) Ratio of the average square of the distances, suspects vs. true nearest neighbors (see Table 19).



(b) Proportion of suspects among true nearest neighbors (see Table 20).

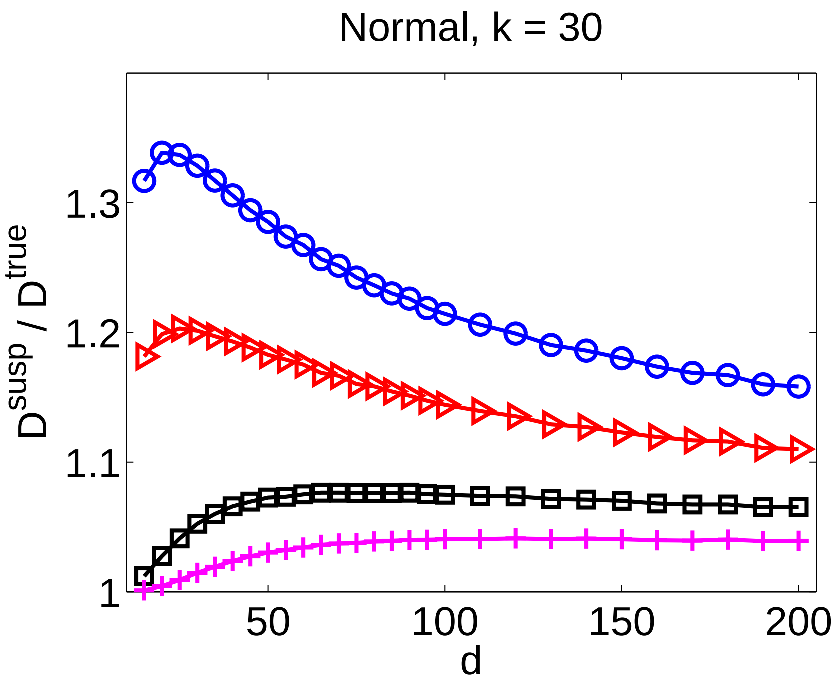
Figure 6: Normal distribution, number of requested nearest neighbors: $k = 15$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d (see Section 5.2.1). RANN parameters: one iteration without supercharging (circles), one iteration with supercharging (triangles), ten iterations without supercharging (squares), ten iterations with supercharging (plus signs).

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.13167E+01	0.10120E+01	0.11813E+01	0.10010E+01
20	0.13384E+01	0.10274E+01	0.11987E+01	0.10043E+01
25	0.13368E+01	0.10412E+01	0.12029E+01	0.10091E+01
30	0.13285E+01	0.10525E+01	0.12013E+01	0.10145E+01
35	0.13171E+01	0.10600E+01	0.11970E+01	0.10193E+01
40	0.13055E+01	0.10658E+01	0.11929E+01	0.10237E+01
45	0.12941E+01	0.10695E+01	0.11883E+01	0.10273E+01
50	0.12852E+01	0.10725E+01	0.11827E+01	0.10302E+01
55	0.12739E+01	0.10732E+01	0.11790E+01	0.10322E+01
60	0.12673E+01	0.10751E+01	0.11751E+01	0.10341E+01
65	0.12564E+01	0.10764E+01	0.11687E+01	0.10362E+01
70	0.12513E+01	0.10763E+01	0.11665E+01	0.10372E+01
75	0.12422E+01	0.10763E+01	0.11601E+01	0.10377E+01
80	0.12363E+01	0.10763E+01	0.11583E+01	0.10388E+01
85	0.12300E+01	0.10762E+01	0.11544E+01	0.10393E+01
90	0.12260E+01	0.10763E+01	0.11512E+01	0.10400E+01
95	0.12187E+01	0.10752E+01	0.11473E+01	0.10400E+01
100	0.12142E+01	0.10748E+01	0.11442E+01	0.10405E+01
110	0.12059E+01	0.10739E+01	0.11394E+01	0.10407E+01
120	0.11990E+01	0.10735E+01	0.11353E+01	0.10411E+01
130	0.11901E+01	0.10716E+01	0.11291E+01	0.10406E+01
140	0.11859E+01	0.10710E+01	0.11270E+01	0.10409E+01
150	0.11801E+01	0.10701E+01	0.11229E+01	0.10405E+01
160	0.11735E+01	0.10681E+01	0.11194E+01	0.10397E+01
170	0.11687E+01	0.10673E+01	0.11168E+01	0.10394E+01
180	0.11671E+01	0.10673E+01	0.11158E+01	0.10402E+01
190	0.11598E+01	0.10652E+01	0.11108E+01	0.10391E+01
200	0.11581E+01	0.10653E+01	0.11099E+01	0.10393E+01

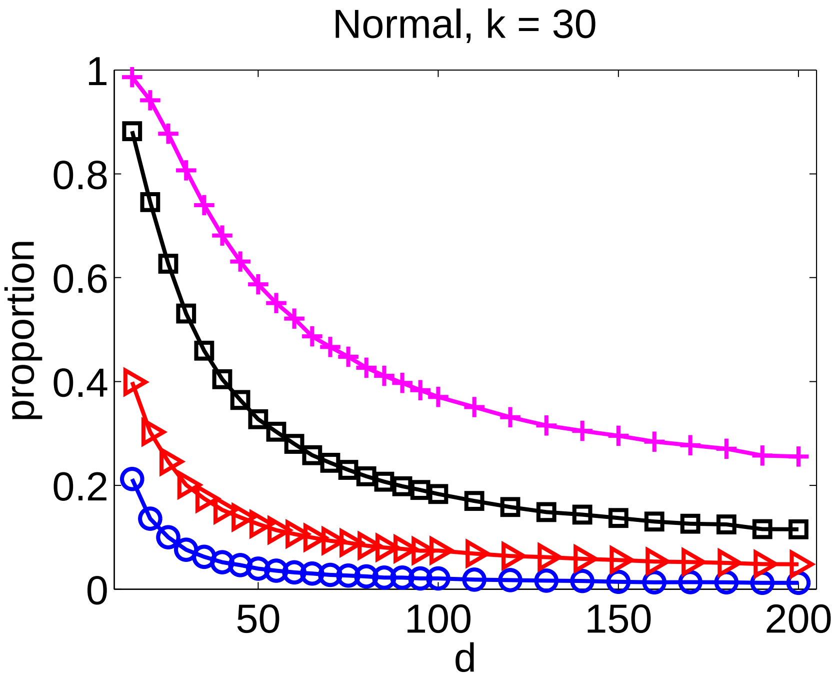
Table 21: Ratio of the average square of the distances, suspects vs. true nearest neighbors. Normal distribution, number of requested nearest neighbors: $k = 30$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 7(a).

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.21207E+00	0.88223E+00	0.39881E+00	0.98602E+00
20	0.13572E+00	0.74548E+00	0.30262E+00	0.94152E+00
25	0.99866E-01	0.62679E+00	0.24563E+00	0.87733E+00
30	0.75826E-01	0.53078E+00	0.20073E+00	0.80652E+00
35	0.62139E-01	0.45932E+00	0.17439E+00	0.73955E+00
40	0.51720E-01	0.40445E+00	0.15264E+00	0.68105E+00
45	0.46233E-01	0.36437E+00	0.13835E+00	0.63105E+00
50	0.39392E-01	0.32730E+00	0.12529E+00	0.58726E+00
55	0.35660E-01	0.30319E+00	0.11382E+00	0.55093E+00
60	0.32226E-01	0.27981E+00	0.10607E+00	0.52110E+00
65	0.30220E-01	0.25810E+00	0.99299E-01	0.48709E+00
70	0.27379E-01	0.24348E+00	0.93126E-01	0.46652E+00
75	0.26160E-01	0.22987E+00	0.89233E-01	0.44777E+00
80	0.24493E-01	0.21738E+00	0.83666E-01	0.42635E+00
85	0.22413E-01	0.20694E+00	0.80293E-01	0.41077E+00
90	0.22299E-01	0.19819E+00	0.77673E-01	0.39710E+00
95	0.20760E-01	0.19105E+00	0.73840E-01	0.38317E+00
100	0.20493E-01	0.18318E+00	0.74219E-01	0.37023E+00
110	0.18493E-01	0.16979E+00	0.68326E-01	0.35069E+00
120	0.17386E-01	0.15835E+00	0.63746E-01	0.33120E+00
130	0.16380E-01	0.14855E+00	0.61486E-01	0.31542E+00
140	0.15733E-01	0.14369E+00	0.58340E-01	0.30502E+00
150	0.14306E-01	0.13722E+00	0.55967E-01	0.29556E+00
160	0.13313E-01	0.13012E+00	0.53113E-01	0.28406E+00
170	0.13526E-01	0.12600E+00	0.52213E-01	0.27732E+00
180	0.13019E-01	0.12455E+00	0.50486E-01	0.27049E+00
190	0.12273E-01	0.11570E+00	0.48353E-01	0.25746E+00
200	0.12186E-01	0.11517E+00	0.47920E-01	0.25544E+00

Table 22: Proportion of suspects among true nearest neighbors. Normal distribution, number of requested nearest neighbors: $k = 30$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 7(b).



(a) Ratio of the average square of the distances, suspects vs. true nearest neighbors (see Table 21).



(b) Proportion of suspects among true nearest neighbors (see Table 22).

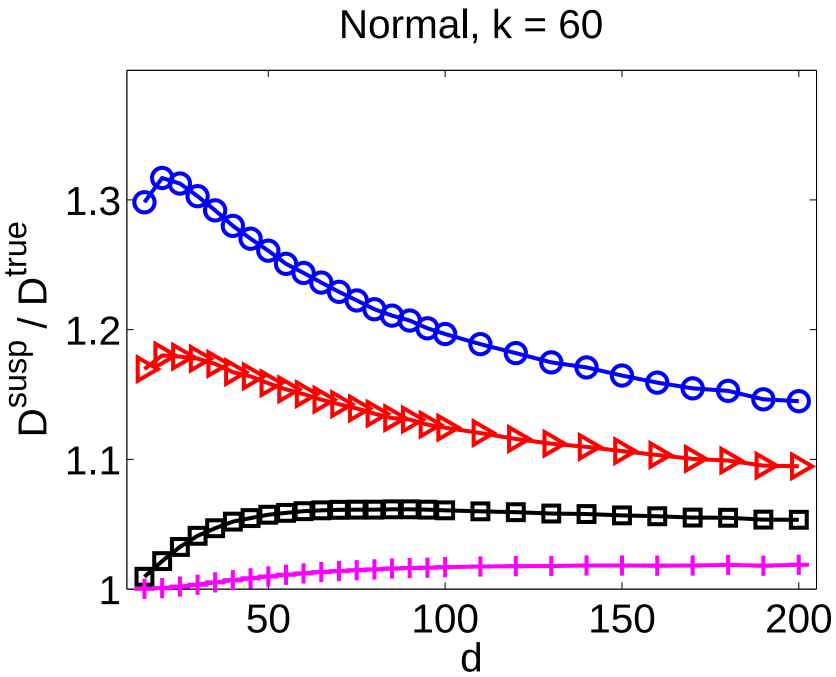
Figure 7: Normal distribution, number of requested nearest neighbors: $k = 30$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d (see Section 5.2.1). RANN parameters: one iteration without supercharging (circles), one iteration with supercharging (triangles), ten iterations without supercharging (squares), ten iterations with supercharging (plus signs).

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.12981E+01	0.10094E+01	0.11696E+01	0.10001E+01
20	0.13169E+01	0.10215E+01	0.11800E+01	0.10007E+01
25	0.13126E+01	0.10324E+01	0.11792E+01	0.10018E+01
30	0.13029E+01	0.10409E+01	0.11775E+01	0.10033E+01
35	0.12919E+01	0.10469E+01	0.11735E+01	0.10051E+01
40	0.12799E+01	0.10519E+01	0.11671E+01	0.10067E+01
45	0.12700E+01	0.10547E+01	0.11639E+01	0.10083E+01
50	0.12608E+01	0.10572E+01	0.11586E+01	0.10096E+01
55	0.12506E+01	0.10587E+01	0.11539E+01	0.10110E+01
60	0.12436E+01	0.10600E+01	0.11503E+01	0.10120E+01
65	0.12363E+01	0.10606E+01	0.11459E+01	0.10131E+01
70	0.12291E+01	0.10610E+01	0.11425E+01	0.10138E+01
75	0.12225E+01	0.10612E+01	0.11392E+01	0.10144E+01
80	0.12157E+01	0.10612E+01	0.11352E+01	0.10151E+01
85	0.12107E+01	0.10614E+01	0.11318E+01	0.10157E+01
90	0.12070E+01	0.10614E+01	0.11306E+01	0.10161E+01
95	0.12010E+01	0.10611E+01	0.11268E+01	0.10164E+01
100	0.11965E+01	0.10606E+01	0.11244E+01	0.10169E+01
110	0.11887E+01	0.10599E+01	0.11202E+01	0.10173E+01
120	0.11819E+01	0.10592E+01	0.11157E+01	0.10175E+01
130	0.11748E+01	0.10582E+01	0.11120E+01	0.10178E+01
140	0.11708E+01	0.10578E+01	0.11096E+01	0.10180E+01
150	0.11646E+01	0.10568E+01	0.11062E+01	0.10181E+01
160	0.11591E+01	0.10561E+01	0.11032E+01	0.10180E+01
170	0.11547E+01	0.10551E+01	0.11003E+01	0.10181E+01
180	0.11527E+01	0.10550E+01	0.10990E+01	0.10185E+01
190	0.11463E+01	0.10534E+01	0.10951E+01	0.10180E+01
200	0.11448E+01	0.10534E+01	0.10946E+01	0.10186E+01

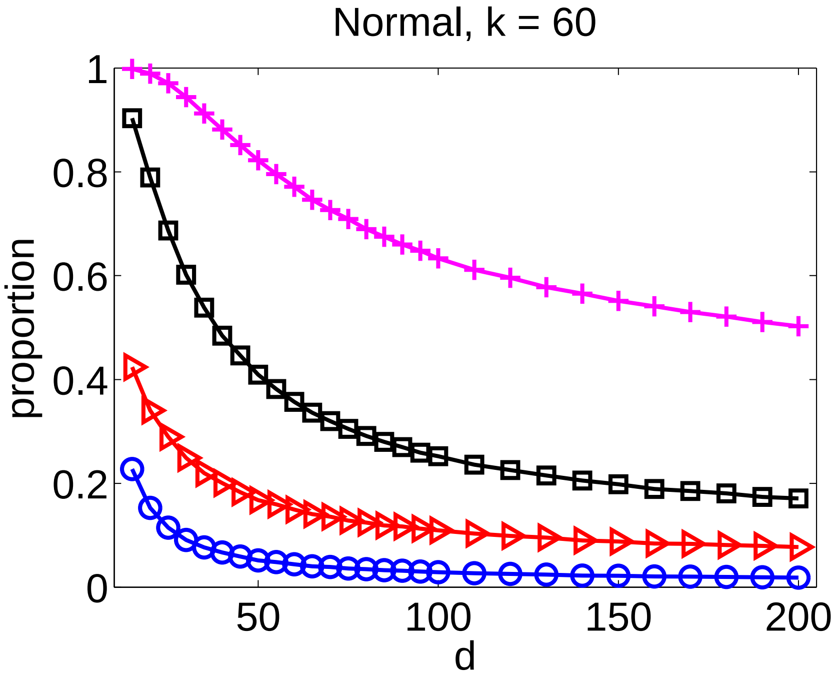
Table 23: Ratio of the average square of the distances, suspects vs. true nearest neighbors. Normal distribution, number of requested nearest neighbors: $k = 60$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 8(a).

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.22758E+00	0.90334E+00	0.42380E+00	0.99811E+00
20	0.15253E+00	0.78899E+00	0.34026E+00	0.98915E+00
25	0.11478E+00	0.68696E+00	0.28945E+00	0.97053E+00
30	0.90873E-01	0.60182E+00	0.24896E+00	0.94376E+00
35	0.76743E-01	0.53831E+00	0.21954E+00	0.91241E+00
40	0.66706E-01	0.48444E+00	0.20054E+00	0.88148E+00
45	0.58966E-01	0.44632E+00	0.18305E+00	0.85145E+00
50	0.51569E-01	0.40913E+00	0.16767E+00	0.82218E+00
55	0.48276E-01	0.38160E+00	0.15935E+00	0.79551E+00
60	0.44100E-01	0.35694E+00	0.14939E+00	0.77118E+00
65	0.40313E-01	0.33599E+00	0.14055E+00	0.74604E+00
70	0.38836E-01	0.31993E+00	0.13582E+00	0.72637E+00
75	0.35919E-01	0.30500E+00	0.12844E+00	0.70909E+00
80	0.34723E-01	0.29134E+00	0.12450E+00	0.68957E+00
85	0.32750E-01	0.27987E+00	0.11882E+00	0.67483E+00
90	0.31580E-01	0.26983E+00	0.11702E+00	0.66010E+00
95	0.29956E-01	0.25901E+00	0.11262E+00	0.64803E+00
100	0.28756E-01	0.25217E+00	0.10932E+00	0.63346E+00
110	0.26743E-01	0.23597E+00	0.10329E+00	0.61133E+00
120	0.25573E-01	0.22566E+00	0.98653E-01	0.59596E+00
130	0.24163E-01	0.21513E+00	0.95543E-01	0.57782E+00
140	0.22403E-01	0.20524E+00	0.89829E-01	0.56545E+00
150	0.21789E-01	0.19827E+00	0.88073E-01	0.55144E+00
160	0.20639E-01	0.18906E+00	0.84066E-01	0.54110E+00
170	0.20356E-01	0.18523E+00	0.83273E-01	0.52985E+00
180	0.19756E-01	0.18071E+00	0.81179E-01	0.52127E+00
190	0.19016E-01	0.17383E+00	0.79460E-01	0.51086E+00
200	0.18303E-01	0.17101E+00	0.77273E-01	0.50253E+00

Table 24: Proportion of suspects among true nearest neighbors. Normal distribution, number of requested nearest neighbors: $k = 60$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 8(b).



(a) Ratio of the average square of the distances, suspects vs. true nearest neighbors (see Table 23).



(b) Proportion of suspects among true nearest neighbors (see Table 24).

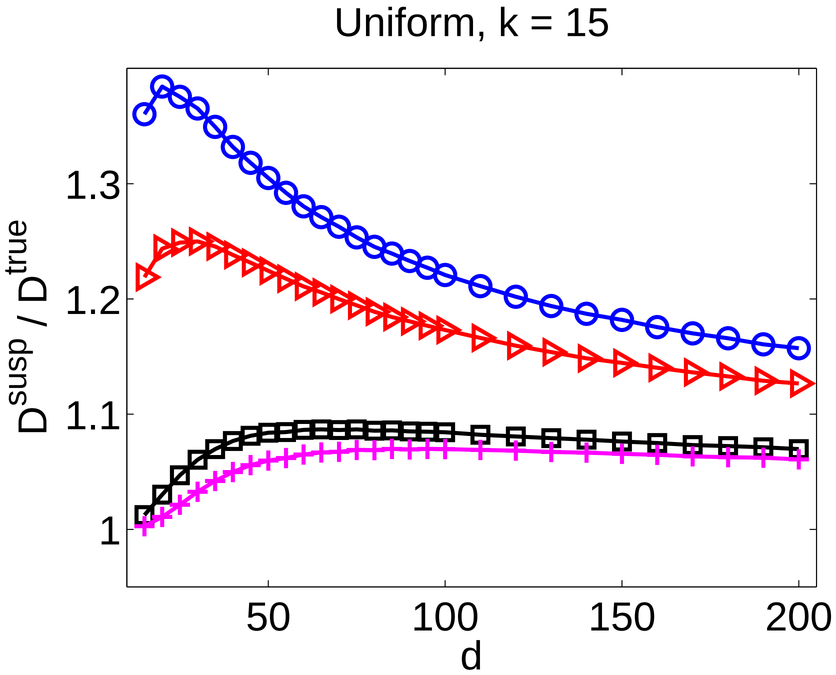
Figure 8: Normal distribution, number of requested nearest neighbors: $k = 60$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d (see Section 5.2.1). RANN parameters: one iteration without supercharging (circles), one iteration with supercharging (triangles), ten iterations without supercharging (squares), ten iterations with supercharging (plus signs).

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.13602E+01	0.10124E+01	0.12187E+01	0.10028E+01
20	0.13842E+01	0.10301E+01	0.12435E+01	0.10108E+01
25	0.13753E+01	0.10469E+01	0.12489E+01	0.10213E+01
30	0.13652E+01	0.10603E+01	0.12498E+01	0.10326E+01
35	0.13493E+01	0.10697E+01	0.12454E+01	0.10419E+01
40	0.13319E+01	0.10767E+01	0.12381E+01	0.10498E+01
45	0.13181E+01	0.10811E+01	0.12314E+01	0.10557E+01
50	0.13049E+01	0.10838E+01	0.12243E+01	0.10596E+01
55	0.12919E+01	0.10845E+01	0.12174E+01	0.10619E+01
60	0.12803E+01	0.10863E+01	0.12107E+01	0.10649E+01
65	0.12709E+01	0.10867E+01	0.12055E+01	0.10667E+01
70	0.12625E+01	0.10861E+01	0.11998E+01	0.10672E+01
75	0.12534E+01	0.10868E+01	0.11944E+01	0.10690E+01
80	0.12451E+01	0.10856E+01	0.11888E+01	0.10688E+01
85	0.12393E+01	0.10859E+01	0.11842E+01	0.10698E+01
90	0.12328E+01	0.10850E+01	0.11804E+01	0.10694E+01
95	0.12270E+01	0.10847E+01	0.11766E+01	0.10698E+01
100	0.12207E+01	0.10843E+01	0.11729E+01	0.10696E+01
110	0.12110E+01	0.10821E+01	0.11660E+01	0.10690E+01
120	0.12018E+01	0.10807E+01	0.11594E+01	0.10683E+01
130	0.11938E+01	0.10791E+01	0.11538E+01	0.10671E+01
140	0.11871E+01	0.10778E+01	0.11484E+01	0.10666E+01
150	0.11818E+01	0.10762E+01	0.11444E+01	0.10655E+01
160	0.11754E+01	0.10749E+01	0.11402E+01	0.10647E+01
170	0.11701E+01	0.10731E+01	0.11362E+01	0.10633E+01
180	0.11658E+01	0.10722E+01	0.11327E+01	0.10626E+01
190	0.11606E+01	0.10712E+01	0.11290E+01	0.10622E+01
200	0.11572E+01	0.10696E+01	0.11266E+01	0.10607E+01

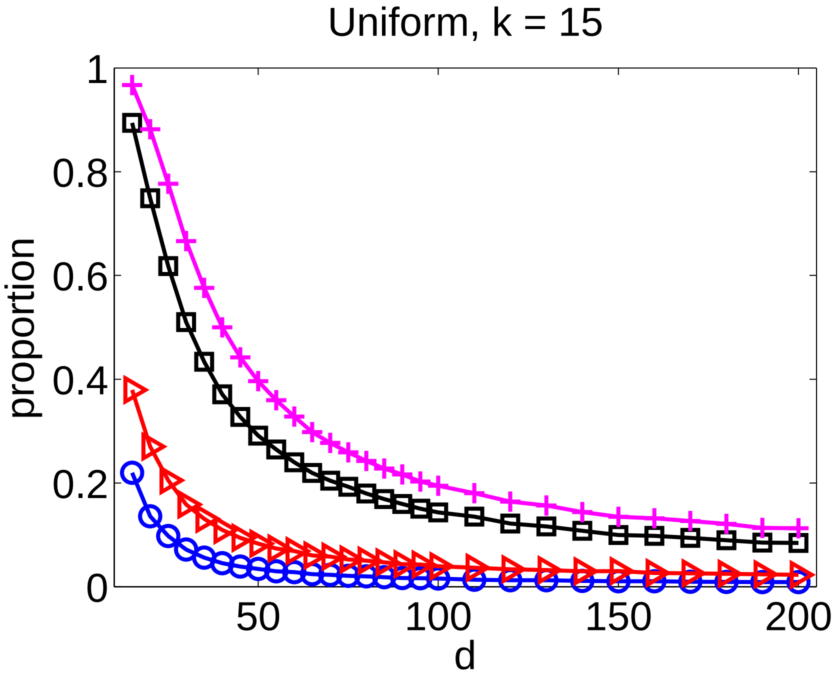
Table 25: Ratio of the average square of the distances, suspects vs. true nearest neighbors. Uniform distribution, number of requested nearest neighbors: $k = 15$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 9(a).

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.21963E+00	0.89414E+00	0.37941E+00	0.96709E+00
20	0.13565E+00	0.74886E+00	0.27003E+00	0.88190E+00
25	0.97599E-01	0.61809E+00	0.20494E+00	0.77673E+00
30	0.71559E-01	0.51005E+00	0.15846E+00	0.66602E+00
35	0.55946E-01	0.43366E+00	0.13094E+00	0.57608E+00
40	0.45493E-01	0.37094E+00	0.10938E+00	0.49994E+00
45	0.38906E-01	0.32725E+00	0.94173E-01	0.44222E+00
50	0.34440E-01	0.29112E+00	0.83240E-01	0.39622E+00
55	0.29853E-01	0.26456E+00	0.73586E-01	0.35941E+00
60	0.27973E-01	0.23967E+00	0.68933E-01	0.32804E+00
65	0.24079E-01	0.21942E+00	0.60186E-01	0.29796E+00
70	0.22786E-01	0.20441E+00	0.57986E-01	0.27711E+00
75	0.20960E-01	0.19267E+00	0.52519E-01	0.25913E+00
80	0.20146E-01	0.17982E+00	0.50186E-01	0.24239E+00
85	0.18466E-01	0.16908E+00	0.47040E-01	0.22773E+00
90	0.16613E-01	0.15963E+00	0.43466E-01	0.21641E+00
95	0.16506E-01	0.15030E+00	0.42613E-01	0.20277E+00
100	0.15879E-01	0.14337E+00	0.39400E-01	0.19478E+00
110	0.13493E-01	0.13502E+00	0.36680E-01	0.18042E+00
120	0.12666E-01	0.12188E+00	0.33639E-01	0.16409E+00
130	0.12653E-01	0.11618E+00	0.31880E-01	0.15666E+00
140	0.11173E-01	0.10767E+00	0.29653E-01	0.14393E+00
150	0.10706E-01	0.99626E-01	0.30026E-01	0.13485E+00
160	0.10466E-01	0.98134E-01	0.26626E-01	0.13199E+00
170	0.94801E-02	0.94093E-01	0.25986E-01	0.12681E+00
180	0.92400E-02	0.89493E-01	0.24839E-01	0.12099E+00
190	0.88133E-02	0.85053E-01	0.23839E-01	0.11330E+00
200	0.88266E-02	0.84306E-01	0.22853E-01	0.11266E+00

Table 26: Proportion of suspects among true nearest neighbors. Uniform distribution, number of requested nearest neighbors: $k = 15$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 9(b).



(a) Ratio of the average square of the distances, suspects vs. true nearest neighbors (see Table 25).



(b) Proportion of suspects among true nearest neighbors (see Table 26).

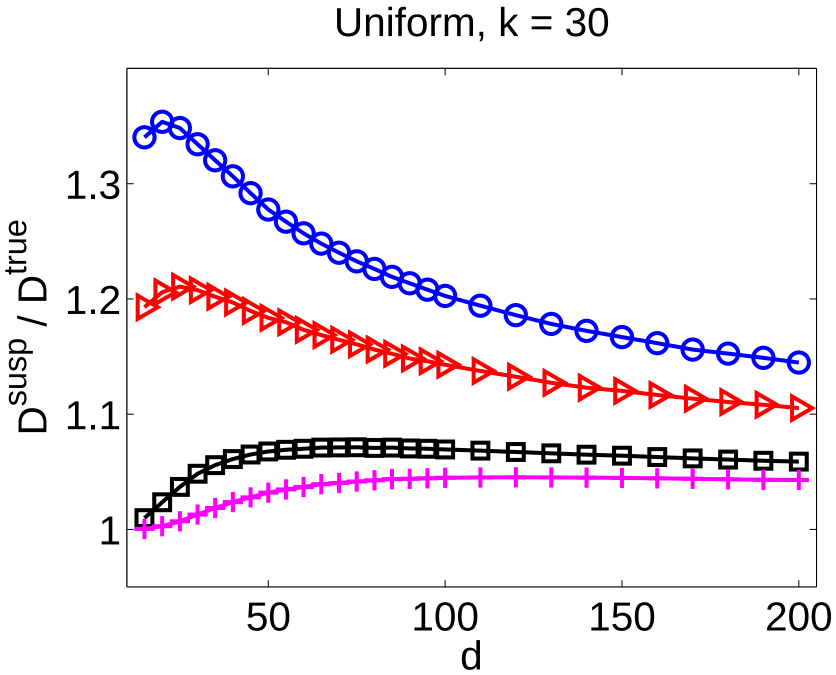
Figure 9: Uniform distribution, number of requested nearest neighbors: $k = 15$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d (see Section 5.2.1). RANN parameters: one iteration without supercharging (circles), one iteration with supercharging (triangles), ten iterations without supercharging (squares), ten iterations with supercharging (plus signs).

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.13401E+01	0.10097E+01	0.11927E+01	0.10005E+01
20	0.13536E+01	0.10236E+01	0.12058E+01	0.10027E+01
25	0.13482E+01	0.10369E+01	0.12106E+01	0.10070E+01
30	0.13342E+01	0.10483E+01	0.12075E+01	0.10129E+01
35	0.13203E+01	0.10557E+01	0.12019E+01	0.10186E+01
40	0.13064E+01	0.10609E+01	0.11968E+01	0.10237E+01
45	0.12917E+01	0.10649E+01	0.11895E+01	0.10278E+01
50	0.12775E+01	0.10676E+01	0.11836E+01	0.10318E+01
55	0.12670E+01	0.10690E+01	0.11798E+01	0.10346E+01
60	0.12569E+01	0.10700E+01	0.11731E+01	0.10367E+01
65	0.12480E+01	0.10709E+01	0.11686E+01	0.10391E+01
70	0.12400E+01	0.10710E+01	0.11645E+01	0.10402E+01
75	0.12325E+01	0.10713E+01	0.11605E+01	0.10416E+01
80	0.12259E+01	0.10707E+01	0.11561E+01	0.10426E+01
85	0.12191E+01	0.10710E+01	0.11523E+01	0.10435E+01
90	0.12135E+01	0.10701E+01	0.11482E+01	0.10438E+01
95	0.12079E+01	0.10699E+01	0.11458E+01	0.10444E+01
100	0.12026E+01	0.10693E+01	0.11428E+01	0.10447E+01
110	0.11940E+01	0.10683E+01	0.11374E+01	0.10450E+01
120	0.11858E+01	0.10672E+01	0.11324E+01	0.10452E+01
130	0.11782E+01	0.10659E+01	0.11271E+01	0.10450E+01
140	0.11722E+01	0.10648E+01	0.11230E+01	0.10448E+01
150	0.11668E+01	0.10639E+01	0.11201E+01	0.10446E+01
160	0.11615E+01	0.10627E+01	0.11165E+01	0.10442E+01
170	0.11561E+01	0.10616E+01	0.11134E+01	0.10439E+01
180	0.11524E+01	0.10606E+01	0.11104E+01	0.10434E+01
190	0.11489E+01	0.10596E+01	0.11081E+01	0.10430E+01
200	0.11448E+01	0.10588E+01	0.11052E+01	0.10428E+01

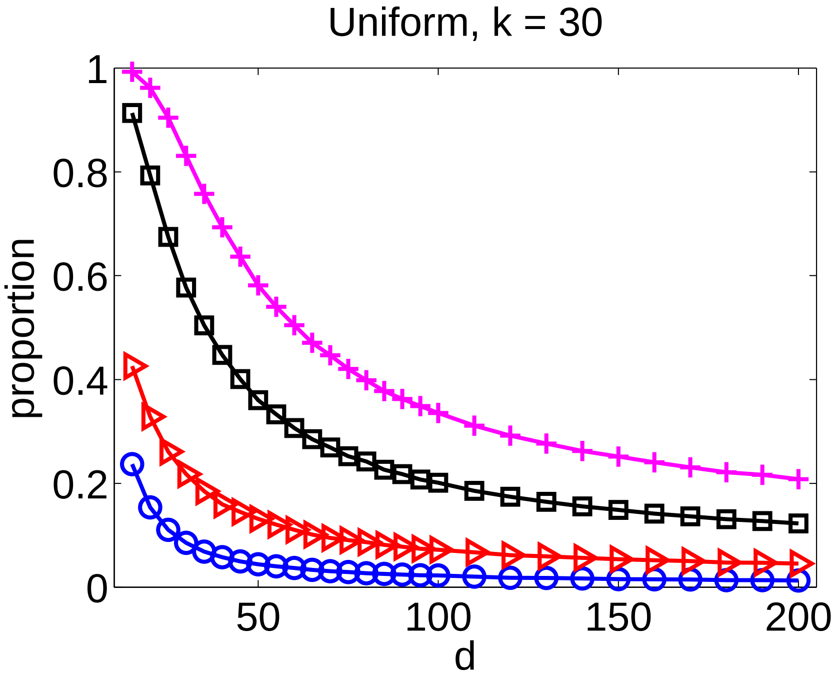
Table 27: Ratio of the average square of the distances, suspects vs. true nearest neighbors. Uniform distribution, number of requested nearest neighbors: $k = 30$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 10(a).

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.23690E+00	0.91333E+00	0.42590E+00	0.99279E+00
20	0.15378E+00	0.79290E+00	0.32813E+00	0.96187E+00
25	0.11056E+00	0.67475E+00	0.26108E+00	0.90432E+00
30	0.85353E-01	0.57711E+00	0.21781E+00	0.83060E+00
35	0.68333E-01	0.50411E+00	0.18499E+00	0.75760E+00
40	0.57880E-01	0.44746E+00	0.15946E+00	0.69308E+00
45	0.49519E-01	0.40067E+00	0.14476E+00	0.63650E+00
50	0.44033E-01	0.36029E+00	0.13145E+00	0.58142E+00
55	0.40193E-01	0.33279E+00	0.12034E+00	0.53988E+00
60	0.36966E-01	0.30664E+00	0.11026E+00	0.50455E+00
65	0.33259E-01	0.28517E+00	0.10120E+00	0.47082E+00
70	0.30686E-01	0.26895E+00	0.94840E-01	0.44663E+00
75	0.29033E-01	0.25210E+00	0.90633E-01	0.42028E+00
80	0.26986E-01	0.24226E+00	0.86779E-01	0.39861E+00
85	0.25626E-01	0.22633E+00	0.81373E-01	0.37768E+00
90	0.24239E-01	0.21757E+00	0.78013E-01	0.36253E+00
95	0.22813E-01	0.20719E+00	0.73959E-01	0.34872E+00
100	0.22526E-01	0.20111E+00	0.72113E-01	0.33530E+00
110	0.20273E-01	0.18559E+00	0.67133E-01	0.31113E+00
120	0.18133E-01	0.17407E+00	0.61713E-01	0.29200E+00
130	0.17699E-01	0.16452E+00	0.59213E-01	0.27673E+00
140	0.16713E-01	0.15552E+00	0.56500E-01	0.26214E+00
150	0.15419E-01	0.14864E+00	0.53773E-01	0.25161E+00
160	0.15113E-01	0.14161E+00	0.51819E-01	0.24060E+00
170	0.14780E-01	0.13646E+00	0.50253E-01	0.23069E+00
180	0.13420E-01	0.13090E+00	0.47413E-01	0.22143E+00
190	0.13340E-01	0.12743E+00	0.46973E-01	0.21650E+00
200	0.12726E-01	0.12278E+00	0.45366E-01	0.20793E+00

Table 28: Proportion of suspects among true nearest neighbors. Uniform distribution, number of requested nearest neighbors: $k = 30$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 10(b).



(a) Ratio of the average square of the distances, suspects vs. true nearest neighbors (see Table 27).



(b) Proportion of suspects among true nearest neighbors (see Table 28).

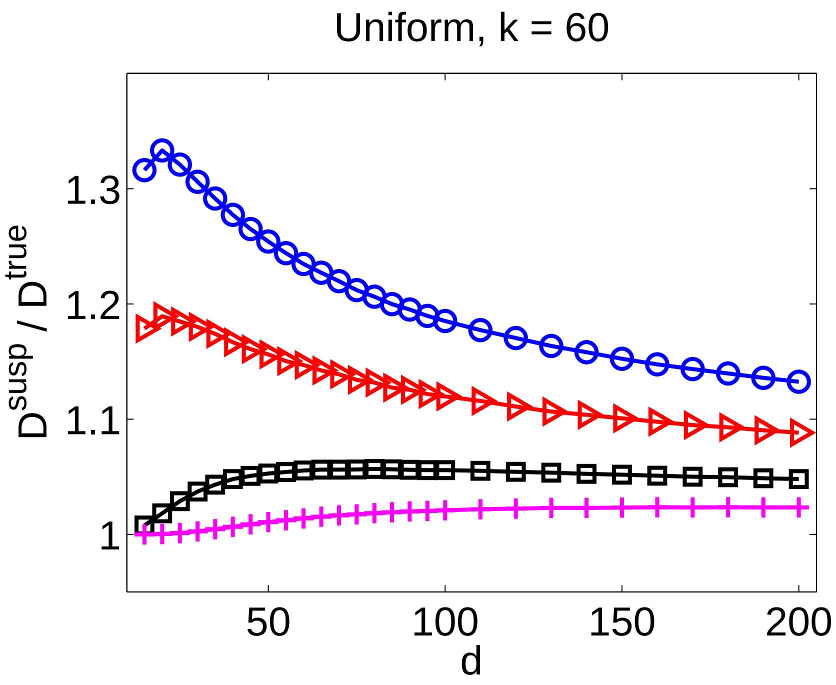
Figure 10: Uniform distribution, number of requested nearest neighbors: $k = 30$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d (see Section 5.2.1). RANN parameters: one iteration without supercharging (circles), one iteration with supercharging (triangles), ten iterations without supercharging (squares), ten iterations with supercharging (plus signs).

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.13160E+01	0.10076E+01	0.11785E+01	0.10000E+01
20	0.13331E+01	0.10182E+01	0.11892E+01	0.10002E+01
25	0.13208E+01	0.10287E+01	0.11847E+01	0.10011E+01
30	0.13059E+01	0.10371E+01	0.11800E+01	0.10026E+01
35	0.12914E+01	0.10432E+01	0.11740E+01	0.10045E+01
40	0.12772E+01	0.10479E+01	0.11667E+01	0.10066E+01
45	0.12649E+01	0.10507E+01	0.11607E+01	0.10087E+01
50	0.12541E+01	0.10528E+01	0.11561E+01	0.10107E+01
55	0.12440E+01	0.10541E+01	0.11503E+01	0.10123E+01
60	0.12347E+01	0.10554E+01	0.11469E+01	0.10139E+01
65	0.12270E+01	0.10562E+01	0.11422E+01	0.10154E+01
70	0.12196E+01	0.10562E+01	0.11389E+01	0.10165E+01
75	0.12119E+01	0.10563E+01	0.11340E+01	0.10174E+01
80	0.12062E+01	0.10567E+01	0.11317E+01	0.10184E+01
85	0.11998E+01	0.10564E+01	0.11274E+01	0.10191E+01
90	0.11951E+01	0.10560E+01	0.11254E+01	0.10198E+01
95	0.11895E+01	0.10557E+01	0.11218E+01	0.10203E+01
100	0.11851E+01	0.10557E+01	0.11197E+01	0.10208E+01
110	0.11772E+01	0.10551E+01	0.11157E+01	0.10218E+01
120	0.11703E+01	0.10543E+01	0.11109E+01	0.10224E+01
130	0.11634E+01	0.10536E+01	0.11065E+01	0.10229E+01
140	0.11580E+01	0.10525E+01	0.11038E+01	0.10230E+01
150	0.11524E+01	0.10517E+01	0.11006E+01	0.10232E+01
160	0.11475E+01	0.10510E+01	0.10977E+01	0.10235E+01
170	0.11434E+01	0.10500E+01	0.10949E+01	0.10234E+01
180	0.11397E+01	0.10495E+01	0.10929E+01	0.10236E+01
190	0.11358E+01	0.10487E+01	0.10902E+01	0.10234E+01
200	0.11325E+01	0.10480E+01	0.10882E+01	0.10234E+01

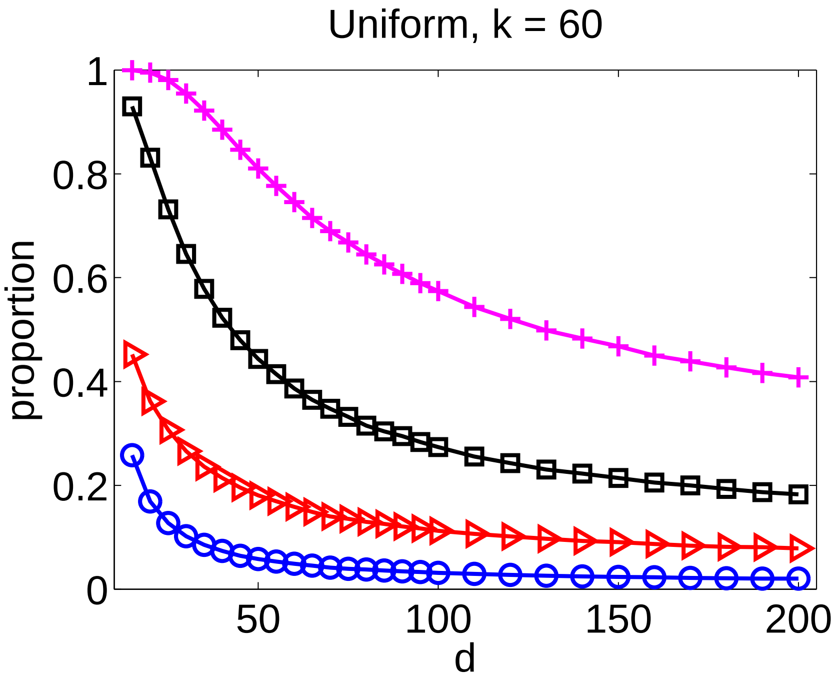
Table 29: Ratio of the average square of the distances, suspects vs. true nearest neighbors. Uniform distribution, number of requested nearest neighbors: $k = 60$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 11(a).

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.25819E+00	0.92979E+00	0.45220E+00	0.99947E+00
20	0.16887E+00	0.83097E+00	0.36171E+00	0.99480E+00
25	0.12723E+00	0.73150E+00	0.30682E+00	0.98069E+00
30	0.10191E+00	0.64583E+00	0.26578E+00	0.95461E+00
35	0.85336E-01	0.57826E+00	0.23609E+00	0.92177E+00
40	0.73763E-01	0.52273E+00	0.21400E+00	0.88508E+00
45	0.64303E-01	0.47955E+00	0.19548E+00	0.84626E+00
50	0.58053E-01	0.44352E+00	0.18057E+00	0.81030E+00
55	0.53079E-01	0.41421E+00	0.16940E+00	0.77684E+00
60	0.48990E-01	0.38665E+00	0.15856E+00	0.74563E+00
65	0.45273E-01	0.36475E+00	0.14861E+00	0.71492E+00
70	0.41559E-01	0.34761E+00	0.14003E+00	0.68969E+00
75	0.39233E-01	0.33180E+00	0.13544E+00	0.66775E+00
80	0.37740E-01	0.31489E+00	0.12955E+00	0.64468E+00
85	0.35990E-01	0.30380E+00	0.12564E+00	0.62537E+00
90	0.34303E-01	0.29489E+00	0.12068E+00	0.60735E+00
95	0.32863E-01	0.28356E+00	0.11678E+00	0.58932E+00
100	0.31479E-01	0.27354E+00	0.11231E+00	0.57407E+00
110	0.29426E-01	0.25551E+00	0.10658E+00	0.54354E+00
120	0.27379E-01	0.24263E+00	0.10167E+00	0.52032E+00
130	0.25893E-01	0.23042E+00	0.97183E-01	0.49840E+00
140	0.24513E-01	0.22274E+00	0.92709E-01	0.48290E+00
150	0.23633E-01	0.21422E+00	0.90410E-01	0.46770E+00
160	0.22929E-01	0.20533E+00	0.86999E-01	0.44988E+00
170	0.21736E-01	0.19974E+00	0.83633E-01	0.43889E+00
180	0.20916E-01	0.19290E+00	0.81413E-01	0.42714E+00
190	0.20360E-01	0.18676E+00	0.80956E-01	0.41630E+00
200	0.20186E-01	0.18259E+00	0.78506E-01	0.40811E+00

Table 30: Proportion of suspects among true nearest neighbors. Uniform distribution, number of requested nearest neighbors: $k = 60$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 11(b).



(a) Ratio of the average square of the distances, suspects vs. true nearest neighbors (see Table 29).



(b) Proportion of suspects among true nearest neighbors (see Table 30).

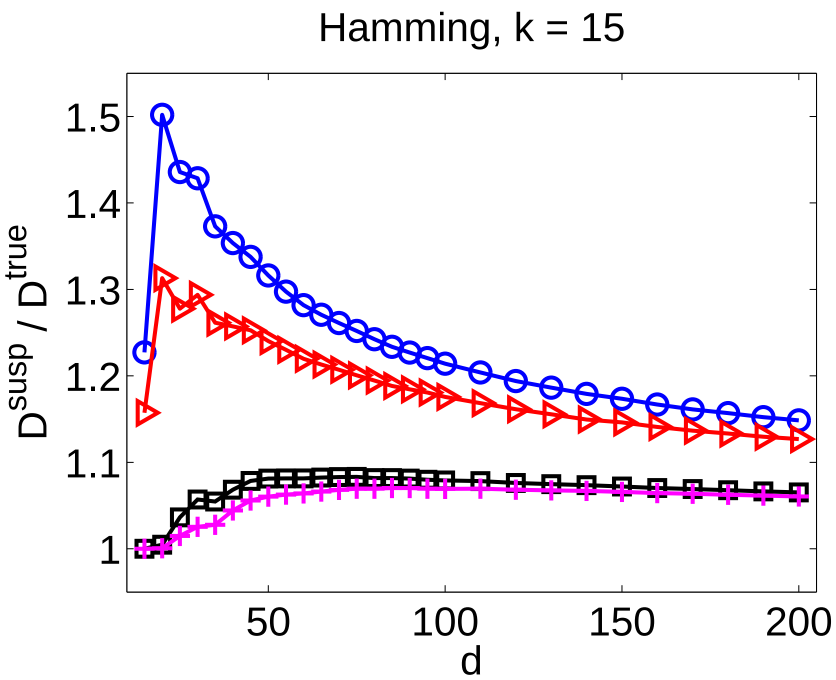
Figure 11: Uniform distribution, number of requested nearest neighbors: $k = 60$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d (see Section 5.2.1). RANN parameters: one iteration without supercharging (circles), one iteration with supercharging (triangles), ten iterations without supercharging (squares), ten iterations with supercharging (plus signs).

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.12272E+01	0.10000E+01	0.11573E+01	0.10000E+01
20	0.15018E+01	0.10046E+01	0.13130E+01	0.10006E+01
25	0.14356E+01	0.10363E+01	0.12776E+01	0.10149E+01
30	0.14286E+01	0.10569E+01	0.12937E+01	0.10254E+01
35	0.13729E+01	0.10546E+01	0.12615E+01	0.10277E+01
40	0.13536E+01	0.10683E+01	0.12573E+01	0.10442E+01
45	0.13375E+01	0.10784E+01	0.12528E+01	0.10558E+01
50	0.13160E+01	0.10812E+01	0.12403E+01	0.10602E+01
55	0.12973E+01	0.10814E+01	0.12301E+01	0.10626E+01
60	0.12817E+01	0.10814E+01	0.12196E+01	0.10641E+01
65	0.12707E+01	0.10823E+01	0.12132E+01	0.10662E+01
70	0.12611E+01	0.10829E+01	0.12071E+01	0.10682E+01
75	0.12518E+01	0.10832E+01	0.12006E+01	0.10695E+01
80	0.12424E+01	0.10817E+01	0.11945E+01	0.10695E+01
85	0.12335E+01	0.10817E+01	0.11884E+01	0.10702E+01
90	0.12269E+01	0.10812E+01	0.11844E+01	0.10703E+01
95	0.12206E+01	0.10798E+01	0.11807E+01	0.10694E+01
100	0.12140E+01	0.10791E+01	0.11757E+01	0.10693E+01
110	0.12039E+01	0.10783E+01	0.11683E+01	0.10694E+01
120	0.11939E+01	0.10762E+01	0.11615E+01	0.10682E+01
130	0.11863E+01	0.10748E+01	0.11556E+01	0.10675E+01
140	0.11790E+01	0.10736E+01	0.11494E+01	0.10669E+01
150	0.11735E+01	0.10719E+01	0.11461E+01	0.10657E+01
160	0.11669E+01	0.10702E+01	0.11409E+01	0.10644E+01
170	0.11613E+01	0.10690E+01	0.11366E+01	0.10636E+01
180	0.11568E+01	0.10677E+01	0.11332E+01	0.10625E+01
190	0.11522E+01	0.10662E+01	0.11296E+01	0.10615E+01
200	0.11484E+01	0.10652E+01	0.11269E+01	0.10605E+01

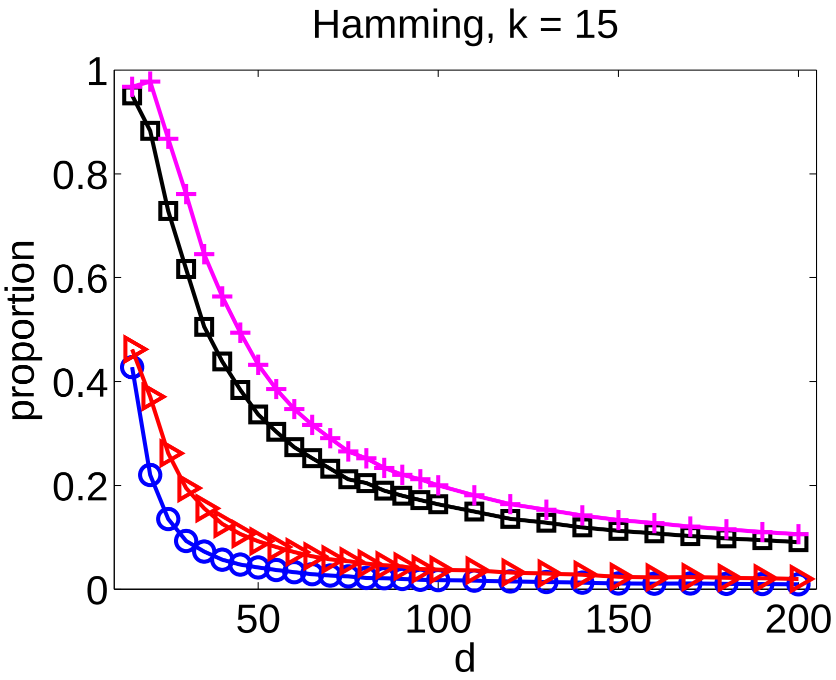
Table 31: Ratio of the average square of the distances, suspects vs. true nearest neighbors. Hamming distribution, number of requested nearest neighbors: $k = 15$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 12(a).

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.42762E+00	0.95102E+00	0.46187E+00	0.96768E+00
20	0.22001E+00	0.88266E+00	0.37069E+00	0.97772E+00
25	0.13504E+00	0.72813E+00	0.26162E+00	0.86738E+00
30	0.92893E-01	0.61641E+00	0.19458E+00	0.76070E+00
35	0.72440E-01	0.50552E+00	0.15573E+00	0.64512E+00
40	0.56853E-01	0.43859E+00	0.12557E+00	0.56389E+00
45	0.47306E-01	0.38378E+00	0.10618E+00	0.49428E+00
50	0.41760E-01	0.33640E+00	0.92053E-01	0.43224E+00
55	0.36893E-01	0.30325E+00	0.80999E-01	0.38524E+00
60	0.32773E-01	0.27349E+00	0.71386E-01	0.34686E+00
65	0.28826E-01	0.25217E+00	0.63399E-01	0.31674E+00
70	0.26160E-01	0.23190E+00	0.57453E-01	0.29044E+00
75	0.24546E-01	0.21157E+00	0.53813E-01	0.26524E+00
80	0.21493E-01	0.20434E+00	0.49106E-01	0.25198E+00
85	0.21133E-01	0.18975E+00	0.46053E-01	0.23351E+00
90	0.19546E-01	0.17999E+00	0.43613E-01	0.22048E+00
95	0.18093E-01	0.17170E+00	0.38986E-01	0.21141E+00
100	0.17439E-01	0.16328E+00	0.37653E-01	0.19978E+00
110	0.16466E-01	0.14950E+00	0.35906E-01	0.18053E+00
120	0.14853E-01	0.13535E+00	0.31933E-01	0.16377E+00
130	0.13799E-01	0.12784E+00	0.30213E-01	0.15315E+00
140	0.12466E-01	0.11889E+00	0.27706E-01	0.14194E+00
150	0.10960E-01	0.11229E+00	0.23973E-01	0.13329E+00
160	0.10706E-01	0.10750E+00	0.22693E-01	0.12717E+00
170	0.11000E-01	0.10217E+00	0.23359E-01	0.12042E+00
180	0.10240E-01	0.97866E-01	0.21853E-01	0.11515E+00
190	0.99733E-02	0.94533E-01	0.20933E-01	0.11018E+00
200	0.94000E-02	0.90613E-01	0.19666E-01	0.10590E+00

Table 32: Proportion of suspects among true nearest neighbors. Hamming distribution, number of requested nearest neighbors: $k = 15$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 12(b).



(a) Ratio of the average square of the distances, suspects vs. true nearest neighbors (see Table 31).



(b) Proportion of suspects among true nearest neighbors (see Table 32).

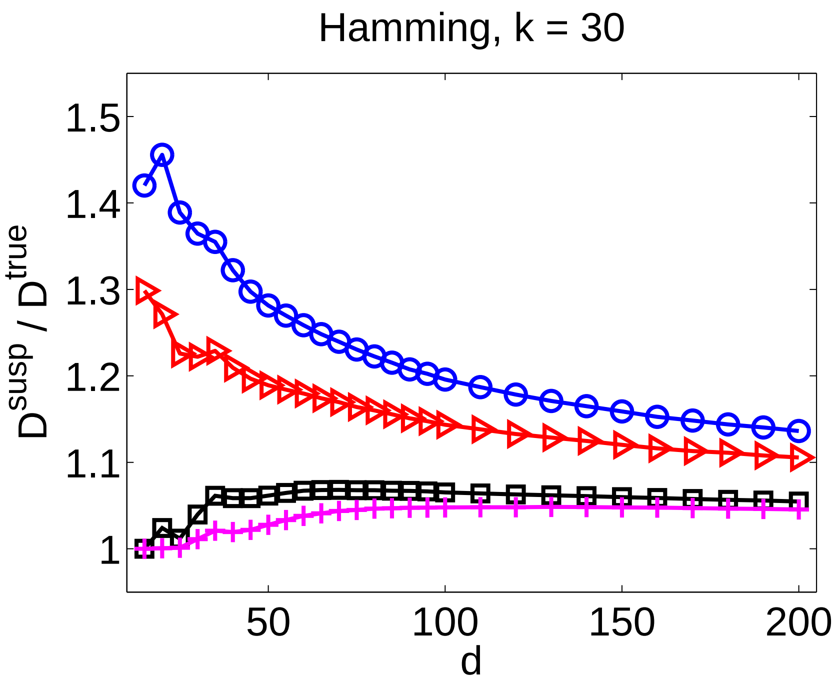
Figure 12: Hamming distribution, number of requested nearest neighbors: $k = 15$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d (see Section 5.2.1). RANN parameters: one iteration without supercharging (circles), one iteration with supercharging (triangles), ten iterations without supercharging (squares), ten iterations with supercharging (plus signs).

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.14201E+01	0.10000E+01	0.12986E+01	0.10000E+01
20	0.14556E+01	0.10239E+01	0.12712E+01	0.10006E+01
25	0.13889E+01	0.10117E+01	0.12258E+01	0.10012E+01
30	0.13645E+01	0.10396E+01	0.12220E+01	0.10111E+01
35	0.13550E+01	0.10613E+01	0.12289E+01	0.10209E+01
40	0.13222E+01	0.10586E+01	0.12093E+01	0.10193E+01
45	0.12973E+01	0.10586E+01	0.11967E+01	0.10218E+01
50	0.12814E+01	0.10615E+01	0.11891E+01	0.10276E+01
55	0.12696E+01	0.10646E+01	0.11839E+01	0.10332E+01
60	0.12584E+01	0.10671E+01	0.11795E+01	0.10380E+01
65	0.12482E+01	0.10680E+01	0.11741E+01	0.10410E+01
70	0.12394E+01	0.10681E+01	0.11696E+01	0.10437E+01
75	0.12304E+01	0.10678E+01	0.11639E+01	0.10449E+01
80	0.12224E+01	0.10677E+01	0.11599E+01	0.10465E+01
85	0.12153E+01	0.10670E+01	0.11555E+01	0.10469E+01
90	0.12075E+01	0.10669E+01	0.11509E+01	0.10475E+01
95	0.12023E+01	0.10664E+01	0.11476E+01	0.10476E+01
100	0.11958E+01	0.10653E+01	0.11434E+01	0.10477E+01
110	0.11870E+01	0.10640E+01	0.11381E+01	0.10480E+01
120	0.11784E+01	0.10628E+01	0.11326E+01	0.10480E+01
130	0.11709E+01	0.10619E+01	0.11285E+01	0.10483E+01
140	0.11648E+01	0.10608E+01	0.11246E+01	0.10482E+01
150	0.11588E+01	0.10598E+01	0.11202E+01	0.10478E+01
160	0.11526E+01	0.10588E+01	0.11161E+01	0.10476E+01
170	0.11483E+01	0.10575E+01	0.11129E+01	0.10470E+01
180	0.11439E+01	0.10566E+01	0.11110E+01	0.10464E+01
190	0.11402E+01	0.10557E+01	0.11079E+01	0.10461E+01
200	0.11361E+01	0.10547E+01	0.11055E+01	0.10456E+01

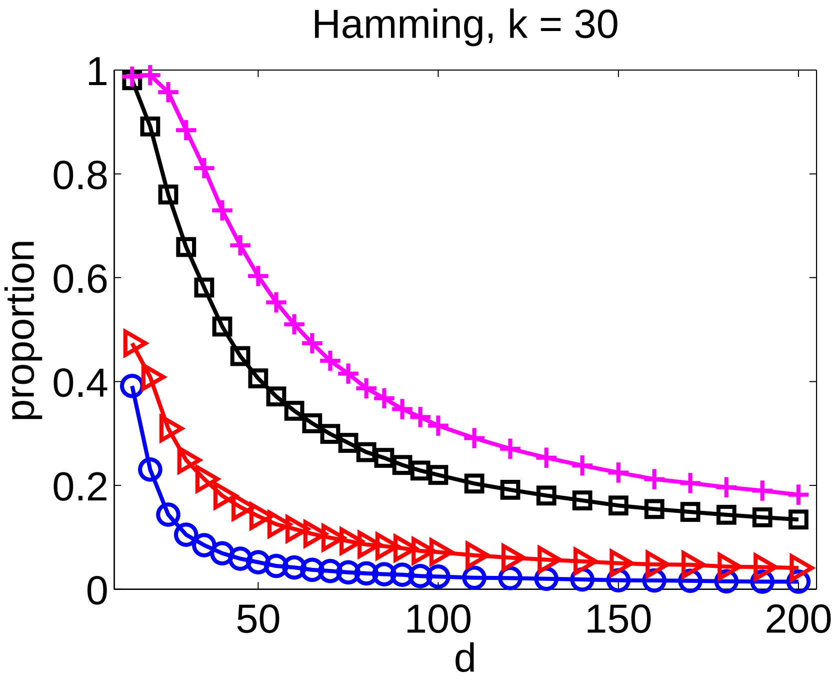
Table 33: Ratio of the average square of the distances, suspects vs. true nearest neighbors. Hamming distribution, number of requested nearest neighbors: $k = 30$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 13(a).

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.39150E+00	0.98028E+00	0.47364E+00	0.98718E+00
20	0.23034E+00	0.89128E+00	0.40849E+00	0.99021E+00
25	0.14365E+00	0.76024E+00	0.30935E+00	0.95715E+00
30	0.10472E+00	0.65917E+00	0.24867E+00	0.88416E+00
35	0.84793E-01	0.58089E+00	0.21203E+00	0.81077E+00
40	0.69053E-01	0.50588E+00	0.18011E+00	0.72949E+00
45	0.58840E-01	0.44904E+00	0.15807E+00	0.66226E+00
50	0.51780E-01	0.40606E+00	0.14081E+00	0.60312E+00
55	0.44839E-01	0.37124E+00	0.12519E+00	0.55248E+00
60	0.41813E-01	0.34367E+00	0.11579E+00	0.51006E+00
65	0.37219E-01	0.31936E+00	0.10640E+00	0.47377E+00
70	0.35046E-01	0.29908E+00	0.98931E-01	0.43997E+00
75	0.32220E-01	0.28192E+00	0.92480E-01	0.41510E+00
80	0.30320E-01	0.26398E+00	0.85993E-01	0.38689E+00
85	0.28866E-01	0.25292E+00	0.82899E-01	0.36759E+00
90	0.27780E-01	0.23930E+00	0.78759E-01	0.34696E+00
95	0.25093E-01	0.22834E+00	0.73560E-01	0.33166E+00
100	0.24173E-01	0.21997E+00	0.71366E-01	0.31498E+00
110	0.22093E-01	0.20336E+00	0.65080E-01	0.29088E+00
120	0.21313E-01	0.19129E+00	0.60753E-01	0.27033E+00
130	0.19906E-01	0.18025E+00	0.56820E-01	0.25303E+00
140	0.18599E-01	0.17076E+00	0.53413E-01	0.23816E+00
150	0.16913E-01	0.16115E+00	0.49893E-01	0.22461E+00
160	0.16613E-01	0.15421E+00	0.47539E-01	0.21189E+00
170	0.16153E-01	0.14867E+00	0.46913E-01	0.20442E+00
180	0.14959E-01	0.14315E+00	0.43479E-01	0.19631E+00
190	0.14506E-01	0.13821E+00	0.42280E-01	0.18984E+00
200	0.14406E-01	0.13403E+00	0.40926E-01	0.18202E+00

Table 34: Proportion of suspects among true nearest neighbors. Hamming distribution, number of requested nearest neighbors: $k = 30$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 13(b).



(a) Ratio of the average square of the distances, suspects vs. true nearest neighbors (see Table 33).



(b) Proportion of suspects among true nearest neighbors (see Table 34).

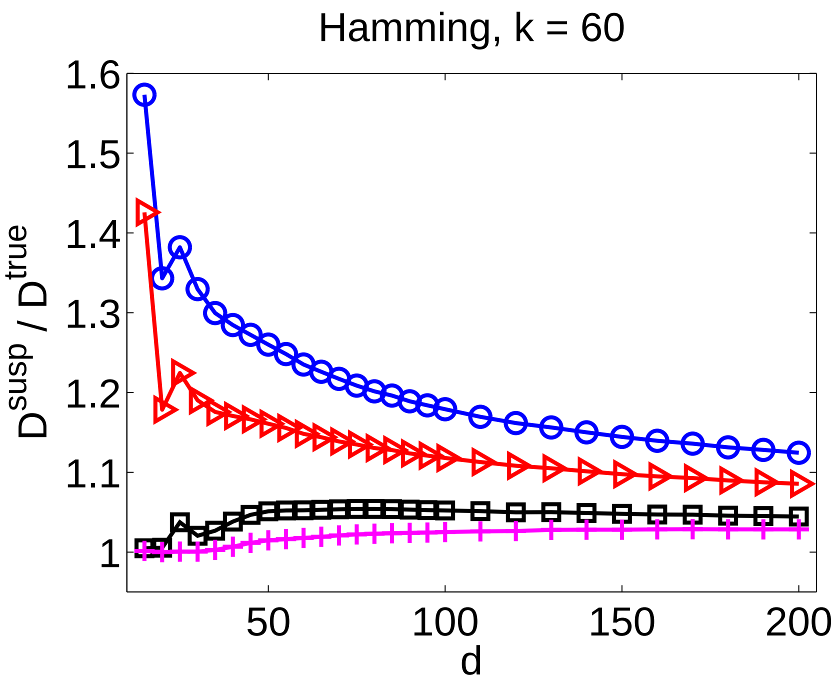
Figure 13: Hamming distribution, number of requested nearest neighbors: $k = 30$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d (see Section 5.2.1). RANN parameters: one iteration without supercharging (circles), one iteration with supercharging (triangles), ten iterations without supercharging (squares), ten iterations with supercharging (plus signs).

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.15732E+01	0.10048E+01	0.14259E+01	0.10015E+01
20	0.13431E+01	0.10055E+01	0.11784E+01	0.10000E+01
25	0.13820E+01	0.10374E+01	0.12244E+01	0.10006E+01
30	0.13295E+01	0.10204E+01	0.11902E+01	0.10005E+01
35	0.12996E+01	0.10268E+01	0.11757E+01	0.10027E+01
40	0.12847E+01	0.10382E+01	0.11707E+01	0.10067E+01
45	0.12723E+01	0.10466E+01	0.11663E+01	0.10115E+01
50	0.12601E+01	0.10512E+01	0.11610E+01	0.10148E+01
55	0.12482E+01	0.10523E+01	0.11556E+01	0.10163E+01
60	0.12351E+01	0.10525E+01	0.11482E+01	0.10178E+01
65	0.12261E+01	0.10530E+01	0.11439E+01	0.10192E+01
70	0.12172E+01	0.10537E+01	0.11386E+01	0.10208E+01
75	0.12087E+01	0.10542E+01	0.11349E+01	0.10221E+01
80	0.12014E+01	0.10541E+01	0.11305E+01	0.10229E+01
85	0.11962E+01	0.10538E+01	0.11280E+01	0.10236E+01
90	0.11890E+01	0.10533E+01	0.11237E+01	0.10240E+01
95	0.11839E+01	0.10525E+01	0.11210E+01	0.10246E+01
100	0.11790E+01	0.10521E+01	0.11179E+01	0.10250E+01
110	0.11695E+01	0.10513E+01	0.11129E+01	0.10260E+01
120	0.11618E+01	0.10500E+01	0.11083E+01	0.10264E+01
130	0.11562E+01	0.10501E+01	0.11051E+01	0.10279E+01
140	0.11502E+01	0.10491E+01	0.11013E+01	0.10282E+01
150	0.11444E+01	0.10480E+01	0.10976E+01	0.10280E+01
160	0.11396E+01	0.10472E+01	0.10948E+01	0.10284E+01
170	0.11359E+01	0.10468E+01	0.10928E+01	0.10287E+01
180	0.11313E+01	0.10457E+01	0.10899E+01	0.10285E+01
190	0.11280E+01	0.10452E+01	0.10874E+01	0.10286E+01
200	0.11245E+01	0.10445E+01	0.10855E+01	0.10285E+01

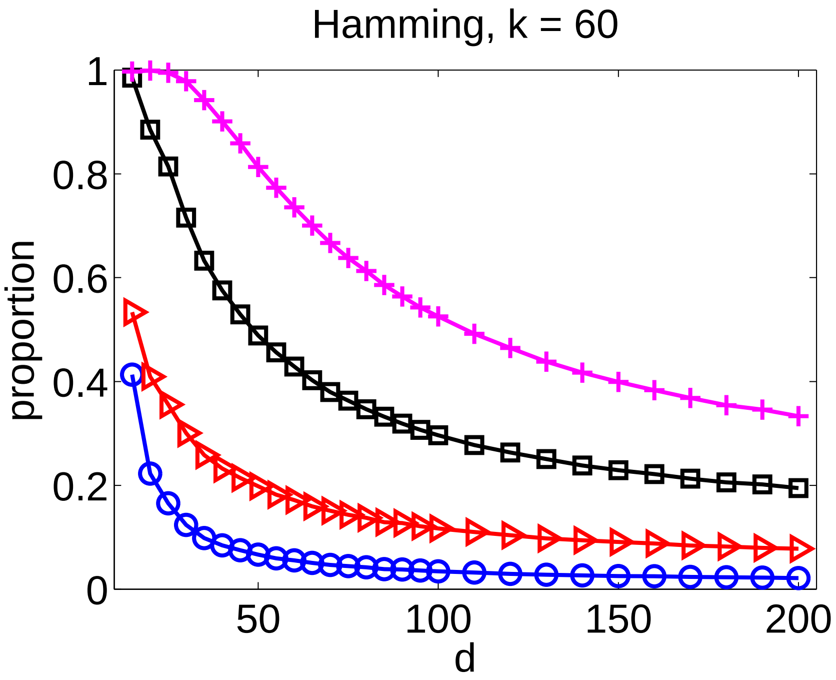
Table 35: Ratio of the average square of the distances, suspects vs. true nearest neighbors. Hamming distribution, number of requested nearest neighbors: $k = 60$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 14(a).

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.41318E+00	0.98505E+00	0.53338E+00	0.99696E+00
20	0.22274E+00	0.88490E+00	0.40937E+00	0.99878E+00
25	0.16532E+00	0.81406E+00	0.35550E+00	0.99457E+00
30	0.12382E+00	0.71557E+00	0.30065E+00	0.97838E+00
35	0.98486E-01	0.63265E+00	0.25833E+00	0.94189E+00
40	0.84280E-01	0.57543E+00	0.23220E+00	0.90106E+00
45	0.74800E-01	0.52942E+00	0.21419E+00	0.85861E+00
50	0.66383E-01	0.48848E+00	0.19832E+00	0.81304E+00
55	0.59393E-01	0.45616E+00	0.18191E+00	0.77301E+00
60	0.55379E-01	0.42870E+00	0.17146E+00	0.73556E+00
65	0.50710E-01	0.40249E+00	0.15935E+00	0.70018E+00
70	0.46719E-01	0.37967E+00	0.15110E+00	0.66678E+00
75	0.44480E-01	0.36307E+00	0.14320E+00	0.63802E+00
80	0.41960E-01	0.34641E+00	0.13751E+00	0.61269E+00
85	0.38536E-01	0.33228E+00	0.12883E+00	0.58599E+00
90	0.37896E-01	0.31899E+00	0.12721E+00	0.56366E+00
95	0.35876E-01	0.30679E+00	0.12101E+00	0.54250E+00
100	0.34416E-01	0.29655E+00	0.11705E+00	0.52529E+00
110	0.32026E-01	0.27749E+00	0.11006E+00	0.49171E+00
120	0.29589E-01	0.26338E+00	0.10390E+00	0.46439E+00
130	0.27699E-01	0.25042E+00	0.97856E-01	0.43819E+00
140	0.26663E-01	0.23817E+00	0.93993E-01	0.41717E+00
150	0.25266E-01	0.22924E+00	0.90890E-01	0.39934E+00
160	0.24896E-01	0.22149E+00	0.88123E-01	0.38320E+00
170	0.23423E-01	0.21284E+00	0.84113E-01	0.36838E+00
180	0.22986E-01	0.20583E+00	0.82203E-01	0.35423E+00
190	0.22259E-01	0.20177E+00	0.79663E-01	0.34573E+00
200	0.21403E-01	0.19474E+00	0.77773E-01	0.33330E+00

Table 36: Proportion of suspects among true nearest neighbors. Hamming distribution, number of requested nearest neighbors: $k = 60$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 14(b).



(a) Ratio of the average square of the distances, suspects vs. true nearest neighbors (see Table 35).



(b) Proportion of suspects among true nearest neighbors (see Table 36).

Figure 14: Hamming distribution, number of requested nearest neighbors: $k = 60$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d (see Section 5.2.1). RANN parameters: one iteration without supercharging (circles), one iteration with supercharging (triangles), ten iterations without supercharging (squares), ten iterations with supercharging (plus signs).

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.16841E+01	0.19641E+02	0.53235E+01	0.23499E+02
20	0.18560E+01	0.21340E+02	0.59924E+01	0.25670E+02
25	0.20328E+01	0.23001E+02	0.63996E+01	0.27706E+02
30	0.21720E+01	0.24441E+02	0.68156E+01	0.29257E+02
35	0.23472E+01	0.26242E+02	0.73429E+01	0.31437E+02
40	0.25098E+01	0.27583E+02	0.76684E+01	0.32923E+02
45	0.26626E+01	0.29383E+02	0.82461E+01	0.35099E+02
50	0.28010E+01	0.30864E+02	0.89230E+01	0.37105E+02
55	0.29762E+01	0.32806E+02	0.95013E+01	0.39222E+02
60	0.31122E+01	0.34600E+02	0.99013E+01	0.41115E+02
65	0.33537E+01	0.36386E+02	0.10378E+02	0.43543E+02
70	0.34786E+01	0.37585E+02	0.10748E+02	0.44958E+02
75	0.36433E+01	0.39345E+02	0.11162E+02	0.47113E+02
80	0.37969E+01	0.40535E+02	0.11525E+02	0.48669E+02
85	0.39474E+01	0.42310E+02	0.11928E+02	0.50706E+02
90	0.40850E+01	0.43683E+02	0.12373E+02	0.52275E+02
95	0.42675E+01	0.45441E+02	0.12763E+02	0.54359E+02
100	0.44034E+01	0.46909E+02	0.13165E+02	0.55826E+02
110	0.47162E+01	0.49976E+02	0.13929E+02	0.59402E+02
120	0.50298E+01	0.53142E+02	0.14754E+02	0.62900E+02
130	0.55699E+01	0.58540E+02	0.16048E+02	0.69020E+02
140	0.58763E+01	0.61584E+02	0.16889E+02	0.72862E+02
150	0.62076E+01	0.64509E+02	0.17671E+02	0.76283E+02
160	0.65787E+01	0.68154E+02	0.18541E+02	0.80897E+02
170	0.68347E+01	0.70952E+02	0.19205E+02	0.84180E+02
180	0.71187E+01	0.73936E+02	0.20049E+02	0.87649E+02
190	0.74629E+01	0.77764E+02	0.20731E+02	0.91211E+02
200	0.77892E+01	0.80588E+02	0.21588E+02	0.95184E+02

Table 37: CPU time of RANN, in seconds. Normal distribution, number of requested nearest neighbors: $k = 15$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 15.

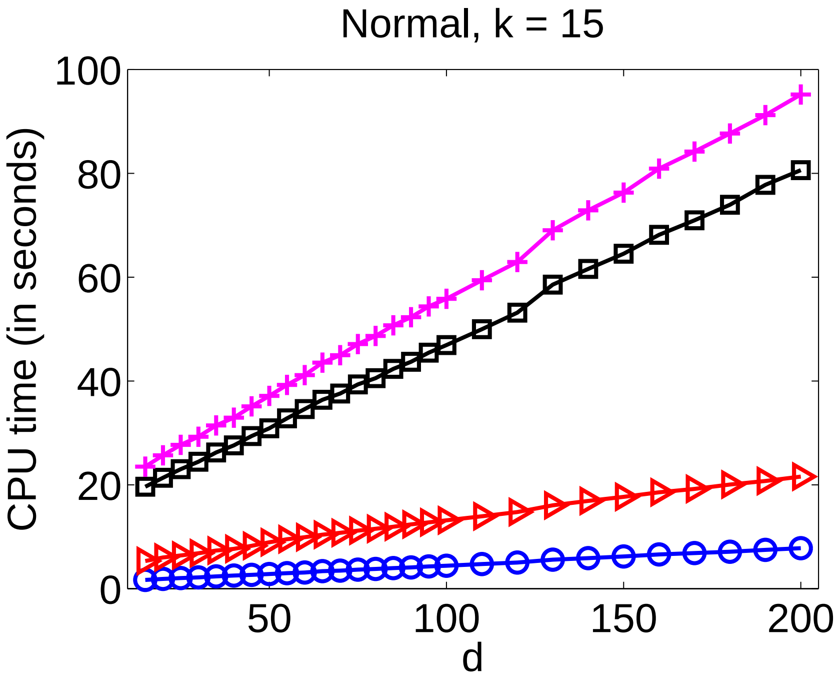


Figure 15: CPU time of RANN, in seconds. Normal distribution, number of requested nearest neighbors: $k = 15$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d (see Section 5.2.1). RANN parameters: one iteration without supercharging (circles), one iteration with supercharging (triangles), ten iterations without supercharging (squares), ten iterations with supercharging (plus signs). See Table 37.

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.16800E+01	0.19554E+02	0.53939E+01	0.23570E+02
20	0.18521E+01	0.21296E+02	0.60388E+01	0.25670E+02
25	0.20240E+01	0.23153E+02	0.65108E+01	0.27934E+02
30	0.21760E+01	0.24474E+02	0.69612E+01	0.29611E+02
35	0.23625E+01	0.26339E+02	0.75244E+01	0.31913E+02
40	0.24929E+01	0.27748E+02	0.81540E+01	0.33944E+02
45	0.26746E+01	0.29378E+02	0.84429E+01	0.35687E+02
50	0.28177E+01	0.30769E+02	0.91421E+01	0.37933E+02
55	0.29817E+01	0.32566E+02	0.96429E+01	0.40222E+02
60	0.31209E+01	0.34278E+02	0.10021E+02	0.41723E+02
65	0.33586E+01	0.36430E+02	0.10501E+02	0.44467E+02
70	0.34705E+01	0.37446E+02	0.10885E+02	0.45834E+02
75	0.36490E+01	0.39160E+02	0.11313E+02	0.48007E+02
80	0.38025E+01	0.40603E+02	0.11726E+02	0.49677E+02
85	0.39649E+01	0.42307E+02	0.12211E+02	0.51402E+02
90	0.40938E+01	0.43682E+02	0.12569E+02	0.53075E+02
95	0.42651E+01	0.45474E+02	0.12966E+02	0.55210E+02
100	0.44154E+01	0.46762E+02	0.13407E+02	0.56916E+02
110	0.47298E+01	0.49882E+02	0.14215E+02	0.60569E+02
120	0.50226E+01	0.53041E+02	0.15076E+02	0.64178E+02
130	0.55803E+01	0.58315E+02	0.16319E+02	0.70387E+02
140	0.58860E+01	0.61530E+02	0.17154E+02	0.74206E+02
150	0.62187E+01	0.64512E+02	0.17693E+02	0.77683E+02
160	0.65763E+01	0.68397E+02	0.18721E+02	0.81965E+02
170	0.68300E+01	0.70902E+02	0.19548E+02	0.84978E+02
180	0.71420E+01	0.73972E+02	0.20314E+02	0.88509E+02
190	0.74572E+01	0.77320E+02	0.21080E+02	0.92459E+02
200	0.78101E+01	0.80741E+02	0.21854E+02	0.96378E+02

Table 38: CPU time of RANN, in seconds. Uniform distribution, number of requested nearest neighbors: $k = 15$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 16.

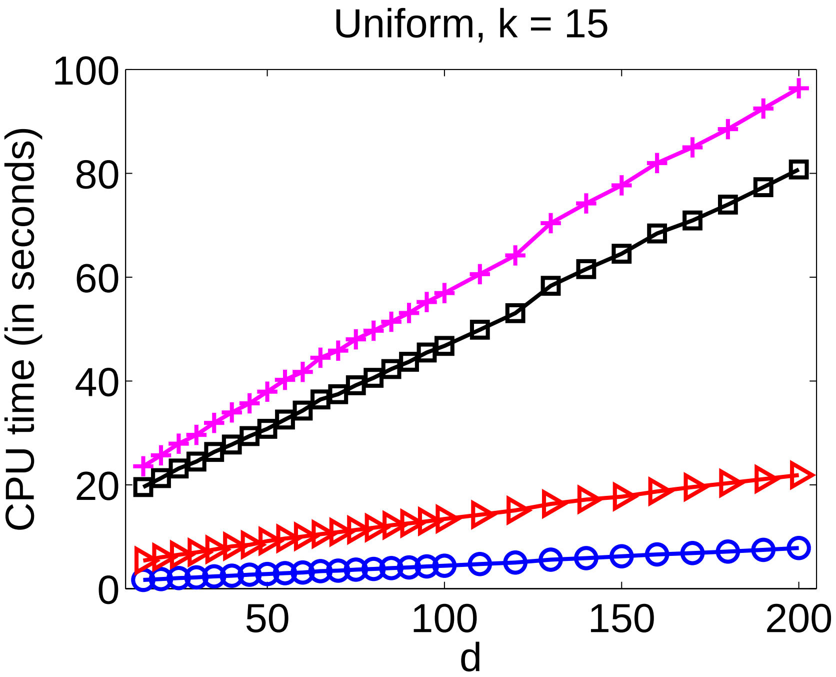


Figure 16: CPU time of RANN, in seconds. Uniform distribution, number of requested nearest neighbors: $k = 15$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d (see Section 5.2.1). RANN parameters: one iteration without supercharging (circles), one iteration with supercharging (triangles), ten iterations without supercharging (squares), ten iterations with supercharging (plus signs). See Table 38.

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.16512E+01	0.19257E+02	0.47027E+01	0.23294E+02
20	0.18409E+01	0.21192E+02	0.58779E+01	0.25666E+02
25	0.20168E+01	0.23051E+02	0.64611E+01	0.27779E+02
30	0.21617E+01	0.24350E+02	0.69212E+01	0.29352E+02
35	0.23633E+01	0.26306E+02	0.74836E+01	0.31671E+02
40	0.25026E+01	0.27616E+02	0.78253E+01	0.33134E+02
45	0.26674E+01	0.29395E+02	0.83901E+01	0.35385E+02
50	0.28097E+01	0.30784E+02	0.91221E+01	0.37393E+02
55	0.29881E+01	0.32604E+02	0.96853E+01	0.39769E+02
60	0.31146E+01	0.33965E+02	0.10155E+02	0.41201E+02
65	0.33570E+01	0.36427E+02	0.10613E+02	0.43955E+02
70	0.34761E+01	0.37530E+02	0.11002E+02	0.45316E+02
75	0.36521E+01	0.39284E+02	0.11409E+02	0.47319E+02
80	0.37898E+01	0.40575E+02	0.11845E+02	0.48776E+02
85	0.39585E+01	0.42398E+02	0.12170E+02	0.50920E+02
90	0.40891E+01	0.44195E+02	0.12552E+02	0.52459E+02
95	0.42698E+01	0.45803E+02	0.12981E+02	0.54579E+02
100	0.44306E+01	0.47323E+02	0.13397E+02	0.56222E+02
110	0.47586E+01	0.50047E+02	0.14154E+02	0.59827E+02
120	0.50715E+01	0.53113E+02	0.15028E+02	0.63369E+02
130	0.55883E+01	0.58432E+02	0.16317E+02	0.69569E+02
140	0.59236E+01	0.61771E+02	0.17180E+02	0.73464E+02
150	0.62507E+01	0.64770E+02	0.17939E+02	0.76823E+02
160	0.66155E+01	0.68563E+02	0.18959E+02	0.81105E+02
170	0.68475E+01	0.71374E+02	0.19497E+02	0.84321E+02
180	0.71403E+01	0.73979E+02	0.20268E+02	0.87701E+02
190	0.74796E+01	0.77415E+02	0.20993E+02	0.91752E+02
200	0.77908E+01	0.80367E+02	0.21886E+02	0.95426E+02

Table 39: CPU time of RANN, in seconds. Hamming distribution, number of requested nearest neighbors: $k = 15$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 17.

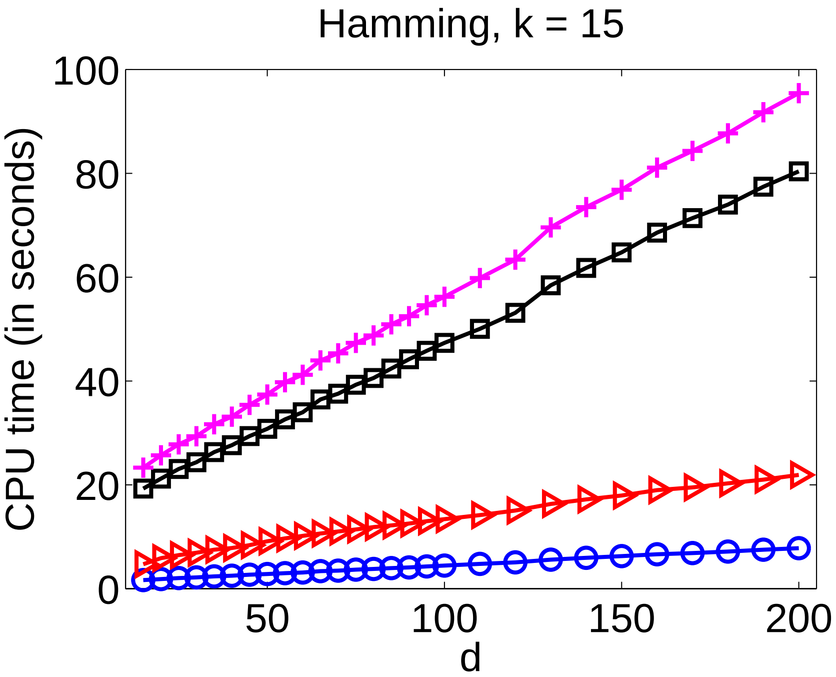


Figure 17: CPU time of RANN, in seconds. Hamming distribution, number of requested nearest neighbors: $k = 15$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d (see Section 5.2.1). RANN parameters: one iteration without supercharging (circles), one iteration with supercharging (triangles), ten iterations without supercharging (squares), ten iterations with supercharging (plus signs). See Table 39.

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.30362E+01	0.36389E+02	0.15610E+02	0.50560E+02
20	0.32753E+01	0.38751E+02	0.17748E+02	0.55101E+02
25	0.35289E+01	0.41325E+02	0.19073E+02	0.58537E+02
30	0.37170E+01	0.43187E+02	0.20312E+02	0.61256E+02
35	0.39890E+01	0.45930E+02	0.21888E+02	0.65618E+02
40	0.41803E+01	0.47827E+02	0.22787E+02	0.68144E+02
45	0.44355E+01	0.50294E+02	0.24597E+02	0.72064E+02
50	0.46274E+01	0.52444E+02	0.26727E+02	0.76611E+02
55	0.48891E+01	0.55008E+02	0.28456E+02	0.80371E+02
60	0.50730E+01	0.56699E+02	0.29494E+02	0.83657E+02
65	0.53971E+01	0.59884E+02	0.30714E+02	0.88326E+02
70	0.55946E+01	0.61648E+02	0.31851E+02	0.90718E+02
75	0.58180E+01	0.64082E+02	0.33000E+02	0.94441E+02
80	0.60507E+01	0.66327E+02	0.34237E+02	0.97410E+02
85	0.62492E+01	0.68626E+02	0.35365E+02	0.10099E+03
90	0.65011E+01	0.70388E+02	0.36562E+02	0.10390E+03
95	0.67035E+01	0.72870E+02	0.37750E+02	0.10808E+03
100	0.70540E+01	0.74795E+02	0.39048E+02	0.11003E+03
110	0.73308E+01	0.79740E+02	0.41098E+02	0.11632E+03
120	0.77685E+01	0.84022E+02	0.43589E+02	0.12250E+03
130	0.85925E+01	0.91645E+02	0.47495E+02	0.13403E+03
140	0.90205E+01	0.96453E+02	0.49945E+02	0.14121E+03
150	0.94486E+01	0.10049E+03	0.52062E+02	0.14682E+03
160	0.99445E+01	0.10555E+03	0.54531E+02	0.15357E+03
170	0.10346E+02	0.10981E+03	0.56565E+02	0.15941E+03
180	0.10785E+02	0.11410E+03	0.58916E+02	0.16558E+03
190	0.11222E+02	0.11858E+03	0.60988E+02	0.17194E+03
200	0.11662E+02	0.12297E+03	0.63433E+02	0.17852E+03

Table 40: CPU time of RANN, in seconds. Normal distribution, number of requested nearest neighbors: $k = 30$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 18.

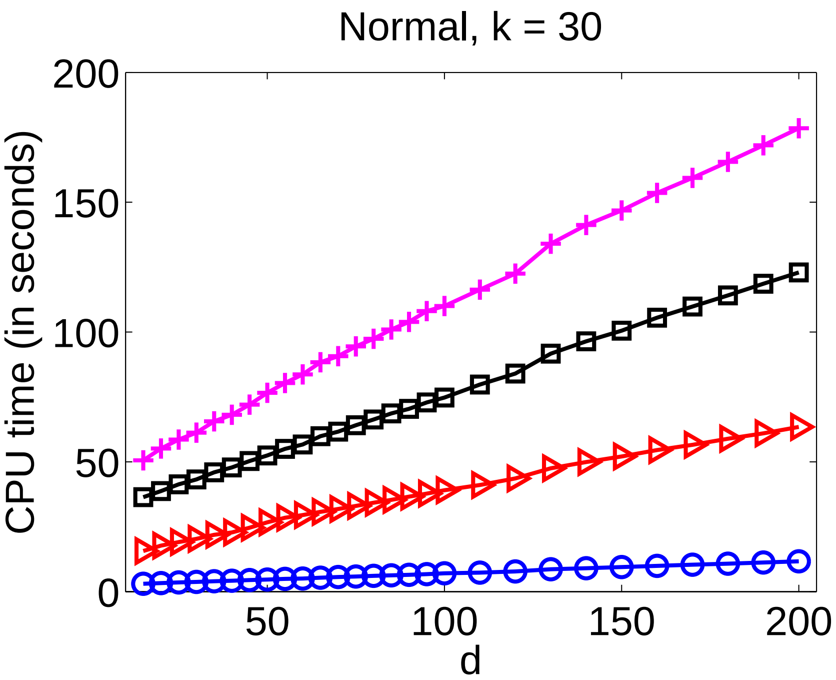


Figure 18: CPU time of RANN, in seconds. Normal distribution, number of requested nearest neighbors: $k = 30$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d (see Section 5.2.1). RANN parameters: one iteration without supercharging (circles), one iteration with supercharging (triangles), ten iterations without supercharging (squares), ten iterations with supercharging (plus signs). See Table 40.

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.30402E+01	0.36516E+02	0.15825E+02	0.51530E+02
20	0.32730E+01	0.39109E+02	0.18000E+02	0.55990E+02
25	0.35354E+01	0.41727E+02	0.19569E+02	0.59828E+02
30	0.37329E+01	0.43465E+02	0.20940E+02	0.62946E+02
35	0.39962E+01	0.46315E+02	0.22693E+02	0.67047E+02
40	0.41939E+01	0.48228E+02	0.24750E+02	0.71141E+02
45	0.44434E+01	0.50661E+02	0.25507E+02	0.74180E+02
50	0.46299E+01	0.52813E+02	0.27855E+02	0.78759E+02
55	0.48986E+01	0.55359E+02	0.29619E+02	0.83082E+02
60	0.51002E+01	0.57049E+02	0.31129E+02	0.86867E+02
65	0.54163E+01	0.60425E+02	0.32379E+02	0.90774E+02
70	0.56011E+01	0.61875E+02	0.33144E+02	0.93301E+02
75	0.58212E+01	0.64433E+02	0.34599E+02	0.97346E+02
80	0.60860E+01	0.66384E+02	0.36398E+02	0.10075E+03
85	0.62668E+01	0.68991E+02	0.37042E+02	0.10395E+03
90	0.64963E+01	0.70667E+02	0.37865E+02	0.10649E+03
95	0.67108E+01	0.74159E+02	0.39160E+02	0.10924E+03
100	0.69428E+01	0.75247E+02	0.40456E+02	0.11261E+03
110	0.73829E+01	0.80046E+02	0.42766E+02	0.11890E+03
120	0.78316E+01	0.84200E+02	0.45339E+02	0.12542E+03
130	0.85804E+01	0.91940E+02	0.49168E+02	0.13647E+03
140	0.90374E+01	0.96741E+02	0.51775E+02	0.14350E+03
150	0.94582E+01	0.10082E+03	0.53883E+02	0.14963E+03
160	0.99951E+01	0.10586E+03	0.56320E+02	0.15644E+03
170	0.10389E+02	0.11049E+03	0.58681E+02	0.16286E+03
180	0.10827E+02	0.11494E+03	0.61298E+02	0.16914E+03
190	0.11241E+02	0.11926E+03	0.63835E+02	0.17507E+03
200	0.11719E+02	0.12427E+03	0.66291E+02	0.18172E+03

Table 41: CPU time of RANN, in seconds. Uniform distribution, number of requested nearest neighbors: $k = 30$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 19.

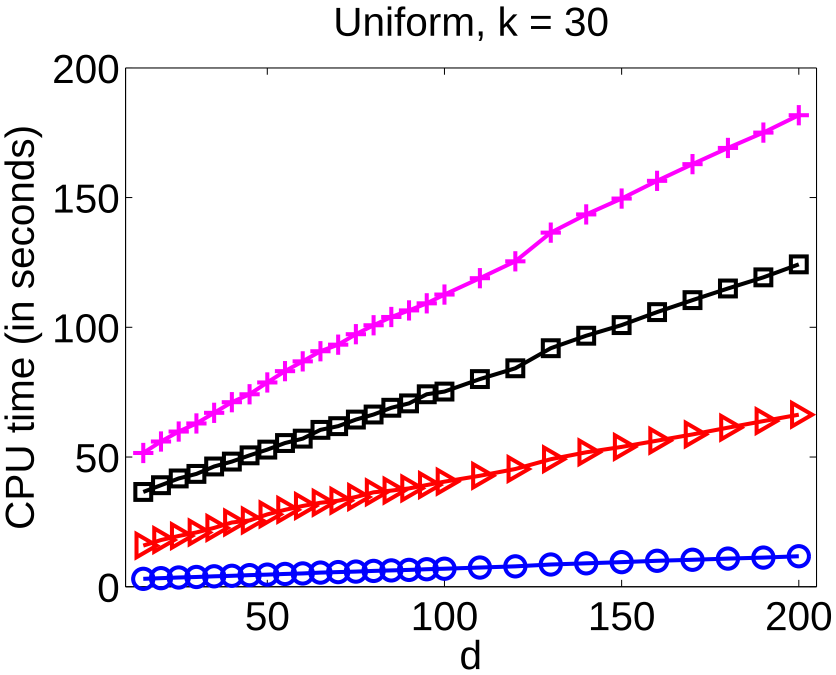


Figure 19: CPU time of RANN, in seconds. Uniform distribution, number of requested nearest neighbors: $k = 30$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d (see Section 5.2.1). RANN parameters: one iteration without supercharging (circles), one iteration with supercharging (triangles), ten iterations without supercharging (squares), ten iterations with supercharging (plus signs). See Table 41.

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.29617E+01	0.35694E+02	0.13571E+02	0.48937E+02
20	0.32522E+01	0.38923E+02	0.17229E+02	0.55115E+02
25	0.35265E+01	0.41273E+02	0.19212E+02	0.59173E+02
30	0.37218E+01	0.43256E+02	0.20744E+02	0.62250E+02
35	0.39882E+01	0.46204E+02	0.22524E+02	0.66387E+02
40	0.41898E+01	0.48176E+02	0.23459E+02	0.69089E+02
45	0.44298E+01	0.50439E+02	0.25386E+02	0.73380E+02
50	0.46283E+01	0.52906E+02	0.27769E+02	0.77932E+02
55	0.48859E+01	0.55009E+02	0.29510E+02	0.82147E+02
60	0.50746E+01	0.56896E+02	0.31103E+02	0.85367E+02
65	0.54346E+01	0.60585E+02	0.32373E+02	0.89628E+02
70	0.55634E+01	0.62415E+02	0.33588E+02	0.92137E+02
75	0.58747E+01	0.64563E+02	0.34787E+02	0.95853E+02
80	0.60012E+01	0.66967E+02	0.36194E+02	0.98881E+02
85	0.63036E+01	0.68689E+02	0.37272E+02	0.10237E+03
90	0.64756E+01	0.70923E+02	0.38310E+02	0.10513E+03
95	0.67444E+01	0.73304E+02	0.39702E+02	0.10859E+03
100	0.69348E+01	0.75297E+02	0.41019E+02	0.11209E+03
110	0.73468E+01	0.80056E+02	0.43354E+02	0.11864E+03
120	0.77956E+01	0.84297E+02	0.45866E+02	0.12497E+03
130	0.85812E+01	0.92396E+02	0.49855E+02	0.13589E+03
140	0.90557E+01	0.97126E+02	0.52439E+02	0.14279E+03
150	0.95045E+01	0.10120E+03	0.54631E+02	0.14897E+03
160	0.99925E+01	0.10623E+03	0.57608E+02	0.15667E+03
170	0.10437E+02	0.11046E+03	0.59366E+02	0.16204E+03
180	0.10881E+02	0.11490E+03	0.61689E+02	0.16828E+03
190	0.11320E+02	0.11896E+03	0.63896E+02	0.17432E+03
200	0.11757E+02	0.12382E+03	0.66527E+02	0.18153E+03

Table 42: CPU time of RANN, in seconds. Hamming distribution, number of requested nearest neighbors: $k = 30$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 20.

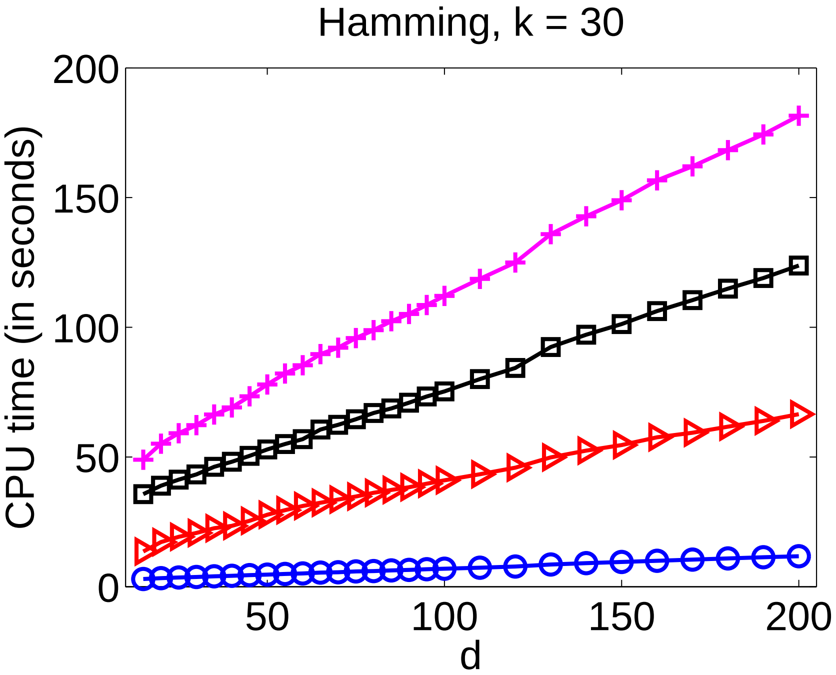


Figure 20: CPU time of RANN, in seconds. Hamming distribution, number of requested nearest neighbors: $k = 30$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d (see Section 5.2.1). RANN parameters: one iteration without supercharging (circles), one iteration with supercharging (triangles), ten iterations without supercharging (squares), ten iterations with supercharging (plus signs). See Table 42.

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.58676E+01	0.72176E+02	0.49055E+02	0.12421E+03
20	0.62324E+01	0.76600E+02	0.56407E+02	0.13571E+03
25	0.66444E+01	0.80075E+02	0.60917E+02	0.14450E+03
30	0.69580E+01	0.83355E+02	0.65214E+02	0.15213E+03
35	0.73668E+01	0.87405E+02	0.70780E+02	0.16092E+03
40	0.76644E+01	0.90187E+02	0.73879E+02	0.16677E+03
45	0.80669E+01	0.94225E+02	0.80164E+02	0.17785E+03
50	0.83533E+01	0.99186E+02	0.87249E+02	0.19041E+03
55	0.87629E+01	0.10122E+03	0.93167E+02	0.19913E+03
60	0.90342E+01	0.10427E+03	0.97772E+02	0.20788E+03
65	0.94982E+01	0.10871E+03	0.10209E+03	0.21586E+03
70	0.97805E+01	0.11135E+03	0.10595E+03	0.22285E+03
75	0.10161E+02	0.11521E+03	0.11038E+03	0.23128E+03
80	0.10463E+02	0.11927E+03	0.11431E+03	0.24070E+03
85	0.10853E+02	0.12214E+03	0.11878E+03	0.24621E+03
90	0.11149E+02	0.12526E+03	0.12294E+03	0.25462E+03
95	0.11593E+02	0.12890E+03	0.12729E+03	0.26190E+03
100	0.11833E+02	0.13155E+03	0.13173E+03	0.26854E+03
110	0.12526E+02	0.13952E+03	0.13923E+03	0.28386E+03
120	0.13219E+02	0.14550E+03	0.14800E+03	0.29806E+03
130	0.14458E+02	0.15834E+03	0.16218E+03	0.32369E+03
140	0.15152E+02	0.16588E+03	0.17150E+03	0.34014E+03
150	0.15820E+02	0.17197E+03	0.18191E+03	0.35363E+03
160	0.16575E+02	0.17894E+03	0.18977E+03	0.36899E+03
170	0.17192E+02	0.18669E+03	0.19551E+03	0.38423E+03
180	0.17900E+02	0.19265E+03	0.20670E+03	0.39853E+03
190	0.18646E+02	0.19982E+03	0.21455E+03	0.41236E+03
200	0.19329E+02	0.20716E+03	0.22096E+03	0.43119E+03

Table 43: CPU time of RANN, in seconds. Normal distribution, number of requested nearest neighbors: $k = 60$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 21.

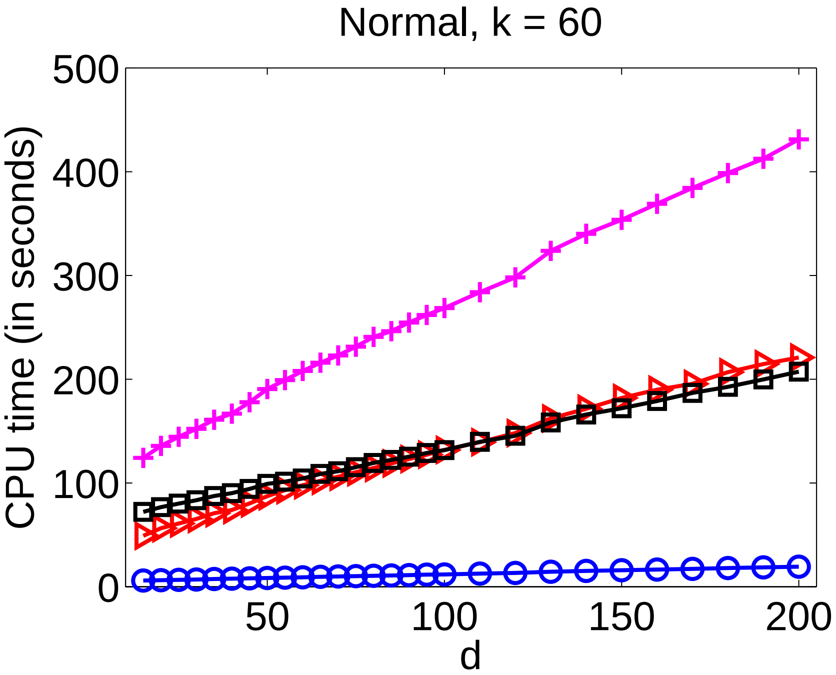


Figure 21: CPU time of RANN, in seconds. Normal distribution, number of requested nearest neighbors: $k = 60$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d (see Section 5.2.1). RANN parameters: one iteration without supercharging (circles), one iteration with supercharging (triangles), ten iterations without supercharging (squares), ten iterations with supercharging (plus signs). See Table 43.

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.58747E+01	0.76515E+02	0.49700E+02	0.12614E+03
20	0.62612E+01	0.79921E+02	0.56952E+02	0.13693E+03
25	0.66924E+01	0.80649E+02	0.62528E+02	0.14635E+03
30	0.69988E+01	0.84404E+02	0.67305E+02	0.15455E+03
35	0.74173E+01	0.88295E+02	0.73479E+02	0.16512E+03
40	0.76989E+01	0.90638E+02	0.80773E+02	0.17640E+03
45	0.81100E+01	0.94445E+02	0.83493E+02	0.18245E+03
50	0.84148E+01	0.98394E+02	0.91506E+02	0.19622E+03
55	0.88294E+01	0.10166E+03	0.97369E+02	0.20582E+03
60	0.91166E+01	0.10501E+03	0.10275E+03	0.21459E+03
65	0.95678E+01	0.10904E+03	0.10743E+03	0.22386E+03
70	0.98661E+01	0.11204E+03	0.11164E+03	0.23056E+03
75	0.10239E+02	0.11556E+03	0.11634E+03	0.23885E+03
80	0.10512E+02	0.11947E+03	0.12134E+03	0.24753E+03
85	0.10958E+02	0.12264E+03	0.12519E+03	0.25461E+03
90	0.11225E+02	0.12646E+03	0.12906E+03	0.26148E+03
95	0.11677E+02	0.13030E+03	0.13403E+03	0.27049E+03
100	0.11929E+02	0.13292E+03	0.13896E+03	0.27779E+03
110	0.12633E+02	0.14056E+03	0.14692E+03	0.29488E+03
120	0.13317E+02	0.14736E+03	0.15700E+03	0.31244E+03
130	0.15007E+02	0.15950E+03	0.16986E+03	0.33985E+03
140	0.15938E+02	0.16711E+03	0.17959E+03	0.35967E+03
150	0.16692E+02	0.17386E+03	0.18797E+03	0.37855E+03
160	0.17480E+02	0.18119E+03	0.19612E+03	0.38305E+03
170	0.18108E+02	0.18900E+03	0.20503E+03	0.40063E+03
180	0.18933E+02	0.19533E+03	0.21354E+03	0.41557E+03
190	0.19673E+02	0.20188E+03	0.22140E+03	0.42458E+03
200	0.20360E+02	0.21025E+03	0.23060E+03	0.44012E+03

Table 44: CPU time of RANN, in seconds. Uniform distribution, number of requested nearest neighbors: $k = 60$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 22.

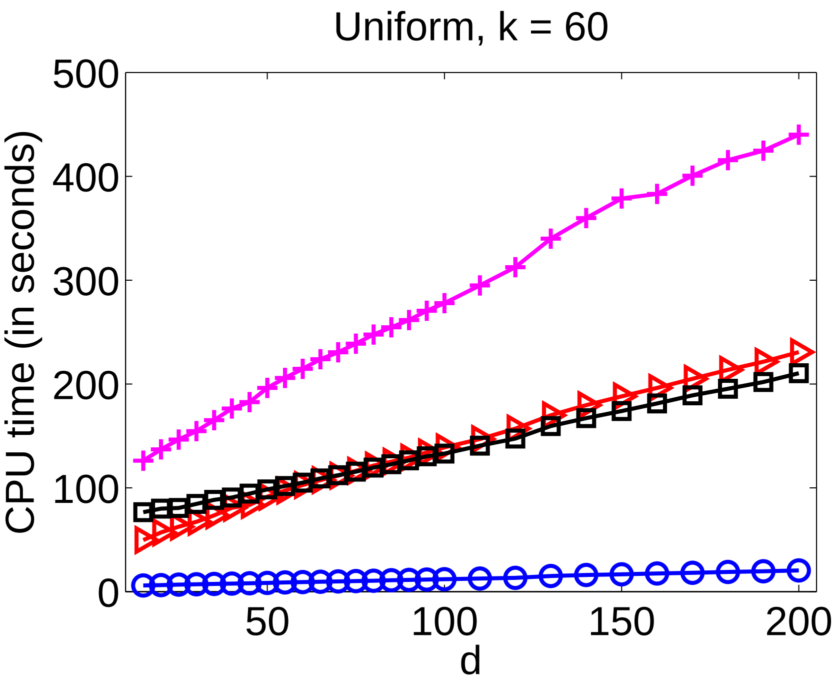


Figure 22: CPU time of RANN, in seconds. Uniform distribution, number of requested nearest neighbors: $k = 60$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d (see Section 5.2.1). RANN parameters: one iteration without supercharging (circles), one iteration with supercharging (triangles), ten iterations without supercharging (squares), ten iterations with supercharging (plus signs). See Table 44.

	without supercharging		with supercharging	
d	$T = 1$	$T = 10$	$T = 1$	$T = 10$
15	0.57156E+01	0.71135E+02	0.44995E+02	0.11480E+03
20	0.62044E+01	0.75629E+02	0.55163E+02	0.13496E+03
25	0.66308E+01	0.80037E+02	0.62056E+02	0.14588E+03
30	0.69628E+01	0.83369E+02	0.67504E+02	0.15416E+03
35	0.73748E+01	0.88315E+02	0.74015E+02	0.16485E+03
40	0.76893E+01	0.90244E+02	0.77513E+02	0.17105E+03
45	0.80757E+01	0.94592E+02	0.84503E+02	0.18265E+03
50	0.83717E+01	0.97469E+02	0.92952E+02	0.19530E+03
55	0.87700E+01	0.10147E+03	0.99125E+02	0.20590E+03
60	0.90806E+01	0.10436E+03	0.10485E+03	0.21473E+03
65	0.95413E+01	0.10989E+03	0.10955E+03	0.22339E+03
70	0.98062E+01	0.11202E+03	0.11387E+03	0.23033E+03
75	0.10248E+02	0.11596E+03	0.11885E+03	0.23848E+03
80	0.10496E+02	0.11834E+03	0.12371E+03	0.24632E+03
85	0.10889E+02	0.12250E+03	0.12780E+03	0.25388E+03
90	0.11184E+02	0.12554E+03	0.13166E+03	0.26092E+03
95	0.11572E+02	0.13013E+03	0.13670E+03	0.26946E+03
100	0.11896E+02	0.13236E+03	0.14170E+03	0.27728E+03
110	0.12561E+02	0.13969E+03	0.14995E+03	0.29227E+03
120	0.13253E+02	0.14637E+03	0.15930E+03	0.30812E+03
130	0.14530E+02	0.15933E+03	0.17376E+03	0.33307E+03
140	0.15226E+02	0.16603E+03	0.18334E+03	0.34908E+03
150	0.15893E+02	0.17305E+03	0.19167E+03	0.36411E+03
160	0.16645E+02	0.18066E+03	0.20232E+03	0.38148E+03
170	0.17271E+02	0.18735E+03	0.20904E+03	0.39435E+03
180	0.17980E+02	0.19464E+03	0.21792E+03	0.40907E+03
190	0.18663E+02	0.20173E+03	0.22593E+03	0.42625E+03
200	0.19358E+02	0.20787E+03	0.23563E+03	0.44319E+03

Table 45: CPU time of RANN, in seconds. Hamming distribution, number of requested nearest neighbors: $k = 60$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d, number of iterations: T (see Section 5.2.1). Corresponds to Figure 23.

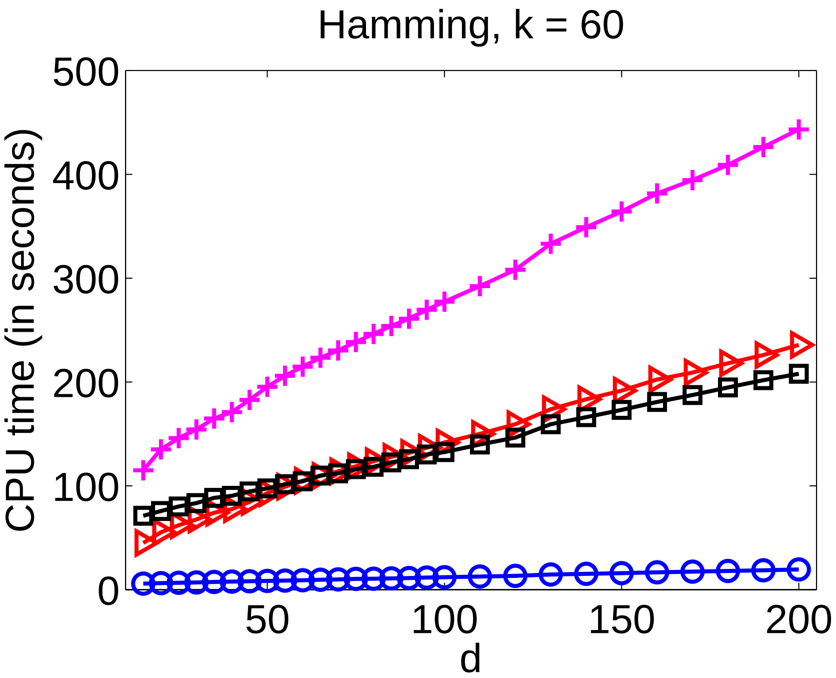


Figure 23: CPU time of RANN, in seconds. Hamming distribution, number of requested nearest neighbors: $k = 60$, number of points: $N = 30 \cdot 2^{12}$, dimensionality: d (see Section 5.2.1). RANN parameters: one iteration without supercharging (circles), one iteration with supercharging (triangles), ten iterations without supercharging (squares), ten iterations with supercharging (plus signs). See Table 45.